WRDC-TR-90-8007
Volume VIII
Part 10

AD-A248 920

||||||||||||||||||||||||||

INTEGRATED INFORMATION SUPPORT SYSTEM (IISS)
Volume VIII - User Interface Subsystem
Part 10 - Graph Support System Unit Test Plan

S. Barker

Control Data Corporation
Integration Technology Services
2970 Presidential Drive
Fairborn, OH   45324-6209

DTIC
ELECTE
APR 22 1992
S   B   D
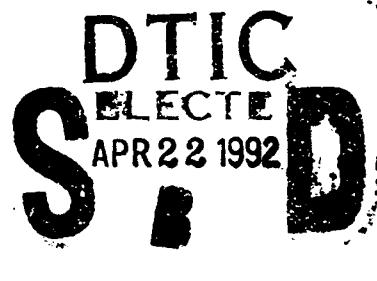
September 1990

Final Report for Period 1 April 1987 - 31 December 1990

92-10245
||||||||||||||||||||||||

MANUFACTURING TECHNOLOGY DIRECTORATE
WRIGHT RESEARCH AND DEVELOPMENT CENTER
AIR FORCE SYSTEMS COMMAND
WRIGHT-PATTERSON AIR FORCE BASE, OHIO   45433-6533

92   4 21 065

NOTICE

When Government drawings, specifications, or other data are used for any purpose other than in connection with a definitely related Government procurement operation, the United States Government thereby incurs no responsibility nor any obligation whatsoever, regardless whether or not the government may have formulated, furnished, or in any way supplied the said drawings, specifications, or other data. It should not, therefore, be construed or implied by any person, persons, or organization that the Government is licensing or conveying any rights or permission to manufacture, use, or market any patented invention that may in any way be related thereto.

This technical report has been reviewed and is approved for publication.
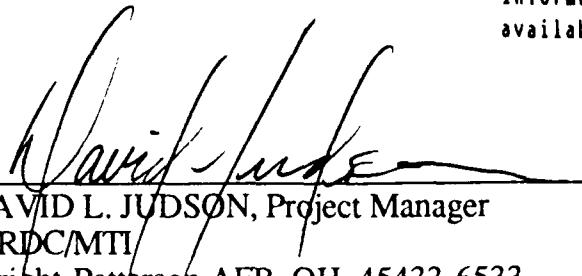
_____
DAVID L. JUDSON, Project Manager
WRDC/MTI
Wright-Patterson AFB, OH 45433-6533

DATE 25 July 91
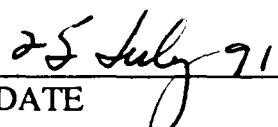
FOR THE COMMANDER:
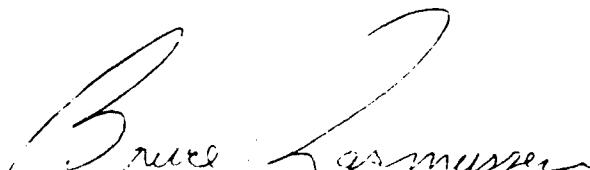
_____
BRUCE A. RASMUSSEN, Chief
WRDC/MTI
Wright-Patterson AFB, OH 45433-6533

DATE 25 July 91

If your address has changed, if you wish to be removed form our mailing list, or if the addressee is no longer employed by your organization please notify WRDC/MTI, Wright-Patterson Air Force Base, OH 45433-6533 to help us maintain a current mailing list.

Copies of this report should not be returned unless return is required by security considerations, contractual obligations, or notice on a specific document.

REPORT DOCUMENTATION PAGE

1a. REPORT SECURITY CLASSIFICATION Unclassified	1b. RESTRICTIVE MARKINGS
2a. SECURITY CLASSIFICATION AUTHORITY	3. DISTRIBUTION/AVAILABILITY OF REPORT Approved for Public Release; Distribution is Unlimited.
2b. DECLASSIFICATION/DOWNGRADING SCHEDULE	

4. PERFORMING ORGANIZATION REPORT NUMBER(S) UTP620344220	5. MONITORING ORGANIZATION REPORT NUMBER(S) WRDC-TR- 90-8007 Vol. VIII, Part 10

6a. NAME OF PERFORMING ORGANIZATION Control Data Corporation; Integration Technology Services	6b. OFFICE SYMBOL (if applicable)	7a. NAME OF MONITORING ORGANIZATION WRDC/MTI

6c. ADDRESS (City,State, and ZIP Code) 2970 Presidential Drive Fairborn, OH 45324-6209	7b. ADDRESS (City, State, and ZIP Code) WPAFB, OH 45433-6533

8a. NAME OF FUNDING/SPONSORING ORGANIZATION Wright Research and Development Center, Air Force Systems Command, USAF	8b. OFFICE SYMBOL (if applicable) WRDC/MTI	9. PROCUREMENT INSTRUMENT IDENTIFICATION NUM. F33600-87-C-0464

8c. ADDRESS (City, State, and ZIP Code)
Wright-Patterson AFB, Ohio 45433-6533

10. SOURCE OF FUNDING NOS.

PROGRAM ELEMENT NO.	PROJECT NO.	TASK NO.	WORK UNIT NO.
78011F	595600	F95600	20950607

11. TITLE   See block 19

12. PERSONAL AUTHOR(S)
Structural Dynamics Research Corporation: Barker, S.

13a. TYPE OF REPORT Final Report	13b. TIME COVERED 4/1/87-12/31/90	14. DATE OF REPORT (Yr.,Mo.,Day) 1990 September 30	15. PAGE COUNT 254

16. SUPPLEMENTARY NOTATION

WRDC/MTI Project Priority 6203

17. COSATI CODES			18. SUBJECT TERMS (Continue on reverse if necessary and identify block no.)
FIELD	GROUP	SUB GR.	
1308	0905		

19. ABSTRACT   (Continue on reverse if necessary and identify block number)

This unit test plan establishes the methodology and procedures used to test the Graph Support System computer program.

BLOCK 11:

INTEGRATED INFORMATION SUPPORT SYSTEM
Vol VIII -User Interface Subsystem

Part 10 - Graph Support System Unit Test Plan

20. DISTRIBUTION/AVAILABILITY OF ABSTRACT UNCLASSIFIED/UNLIMITED x SAME AS RPT. DTIC USERS	21. ABSTRACT SECURITY CLASSIFICATION Unclassified	
22a. NAME OF RESPONSIBLE INDIVIDUAL David L. Judson	22b. TELEPHONE NO. (Include Area Code) (513) 255-7371	22c. OFFICE SYMBOL WRDC/MTI

DD FORM 1473, 83 APR   EDITION OF 1 JAN 73 IS OBSOLETE

SECURITY CLASSIFICATION OF THIS PAGE

FOREWORD

This technical report covers work performed under Air Force
Contract F33600-87-C-0464, DAPro Project. This contract is
sponsored by the Manufacturing Technology Directorate, Air Force
Systems Command, Wright-Patterson Air Force Base, Ohio. It was
administered under the technical direction of Mr. Bruce A.
Rasmussen, Branch Chief, Integration Technology Division,
Manufacturing Technology Directorate, through Mr. David L. Judson,
Project Manager. The Prime Contractor was Integration Technology
Services, Software Programs Division, of the Control Data
Corporation, Dayton, Ohio, under the direction of Mr. W. A.
Osborne. The DAPro Project Manager for Control Data Corporation
was Mr. Jimmy P. Maxwell.

The DAPro project was created to continue the development, test,
and demonstration of the Integrated Information Support System
(IISS). The IISS technology work comprises enhancements to IISS
software and the establishment and operation of IISS test bed
hardware and communications for developers and users.

The following list names the Control Data Corporation
subcontractors and their contributing activities:

SUBCONTRACTOR	ROLE
Control Data Corporation	Responsible for the overall Common Data Model design development and implementation, IISS integration and test, and technology transfer of IISS.
D. Appleton Company	Responsible for providing software information services for the Common Data Model and IDEF1X integration methodology.
ONTEK	Responsible for defining and testing a representative integrated system base in Artificial Intelligence techniques to establish fitness for use.
Simpact Corporation	Responsible for Communication development.
Structural Dynamics Research Corporation	Responsible for User Interfaces, Virtual Terminal Interface, and Network Transaction Manager design, development, implementation, and support.
Arizona State University	Responsible for test bed operations and support.

TABLE OF CONTENTS

LIST OF ILLUSTRATIONS

LIST OF ILLUSTRATIONS (Continued)

LIST OF ILLUSTRATIONS (continued)

LIST OF ILLUSTRATIONS (Continued)

LIST OF ILLUSTRATIONS (Continued)

normalized projection coordinates (NPC): the hypothetical cartesian coordinate system defined within the unit cube (X, Y, and Z coordinates lie between or equal to zero and one). This allows a device independent method for defining a view area.

Page: instance of forms in windows that are created whenever a form is added to a window.

Programmer's Hierarchical Interactive Graphics System (PHIGS): A 2 and 3-dimensional graphics draft standard which is defined independently of any programming language or system.

polyline: a graphics primitive consisting of one or more sequential line segments.

polymarker: a graphics primitive consisting of one or more markers, all of which use the same marker type.

post: to identify a PHIGS structure for display on an open workstation (to display the structure).

redraw all structures: to clear the display surface and then redisplay all previously shown graphics.

structure: a series of elements, transformation definitions, and/or aspects of primitives which define an output image when traversed in a forward direction.

structure network: a set of one or more related structures which combine to form an output display image. Posting of the root structure will display the entire image. The individual structures can not be related in a recursive manor.

transformation: a mathematical conversion which controls the position and orientation of points in space.

text: a geometric primitive consisting of one or more characters. This primitive has the aspects of character height and character color index.

unpost: To remove a structure network from a particular workstation, thereby removing the image from the specified display surface.

update workstation: to display all graphical information which is awaiting display on the display surface of a given workstation.

Window: dynamic area of a terminal screen on which predefined forms may be placed at run time.

workstation: an abstract graphical workstation which provides the logical interface to the applications program. It is analogous to a form in a window.

workstation window: The region in normalized projection coordinates for which graphical output is allowed.

workstation viewport: the true region on the display
surface to which the normalized projection coordinates are
mapped. The units used for this definition are device
specific.

SECTION 2

DEVELOPMENT ACTIVITY

2.1 Statement of Pretest Activity

During system development, the computer programs were tested progressively. Functionality was incrementally tested to verify correct operation of the code. Any bugs which were discovered during this process were corrected as found.

Each form used in the UTP for the Graphics Support System was tested individually. This was conducted by the developer in a manual mode, entering data and then observing the results.

2.2 Pretest Activity Results

Each Graphics Support System callable routine was tested individually. The test program ARTEST was modified to accommodate the calling of the PHIGS C binding routines. This test program allowed the developer to type in the commands and then to immediately observe the results for verification. With this test, all Form Processor calls may be executed. ARTEST is the major testing tool for the Unit Test Plan of the Graphics Support System.

SECTION 3

SYSTEM DESCRIPTION

3.1 System Description

The Graphics Support System is a direct application program interface which allows the generation and manipulation of 3-dimensional graphics information. The system follows the PHIGS standard as closely as possible.

3.2 Testing Schedule

The execution of the Graphic Support System (GSS) is dependent upon the Form Processor, which is in turn dependent upon the NTM subsystem of IISS when it is not configured stand-alone. Testing of the Graphics Support System must be done only after the NTM and Form Processor have been successfully tested. In this unit test plan, the Graphic Support System is dependent upon (and an integral part of) the Form Processor (FP), which is dependent on the Application Interface (AI) and the Virtual Terminal (VT).

3.3 First Location Testing

The tests of the Graphics Support System require the following:

Equipment: Air Force Testbed, Tektronix 4100 series computer terminal.

Support Software: Integrated Information Support System.

Personnel: One integrator familiar with the IISS.

Training: The FP User Manual has been previously provided with the current release.

Deliverables: Current release of the User Interface System

Test Materials: This test may be run interactively by input of the appropriate data and observing the output as outlined in this test plan. A script file has been created to run this unit test plan and save the resulting output.

Security Considerations: None.

3.4 Subsequent Location Testing

The requirements as listed above need to be met; however, in subsequent testing it is advantageous to create a script file of the outlined tests and run this, saving the output of the test for future comparisons.

SECTION 4

TEST SPECIFICATIONS AND EVALUATIONS

4.1 Test Specifications

The Unit Test Plan is designed to cover specific functionality of the Graphic Support system, as outlined in the Form Processor development specification. The divisions of the functional requirements test follow, each of which contains the list of steps during the test (described in Section 5) in which each is used.

A. Controlling the Graphic Support System

1.	open PHIGS	(2)
2.	close PHIGS	(114)
3.	open workstation	(3, 6)
4.	close workstation	(112, 113)
5.	set workstation window 2-D	(4)
6.	set workstation window 3-D	(85)
7.	set workstation viewport 2-D	(5)
8.	set workstation viewport 3-D	(86)

B. Transformation Utilities

9.	scale 2-D	(69)
10.	scale 3-D	(79)
11.	translate 2-D	(66)
12.	translate 3-D	(76)
13.	rotate about X axis	(54)
14.	rotate about Y axis	(60)
15.	rotate about Z axis	(77)
16.	rotate 2-D	(67)
17.	compose matrix 2-D	(68, 70)
18.	compose matrix 3-D	(78, 80)
19.	transform point 2-D	(71, 72, 75)
20.	transform point 3-D	(81, 84)
21.	build transformation matrix 2-D	(73)
22.	build transformation matrix 3-D	(82)
23.	compose transformation matrix 2-D	(74)
24.	compose transformation matrix 3-D	(83)

C. Displaying Messages

25.	message	(103)
26.	error handler	(110)
27.	error log	(111)

D. Controlling Graphic Display

28.	redraw all structures	(101)
29.	update workstation	(50, 51, 53, 102)
30.	post structure	(47, 48, 59, 65, 91, 96, 100)
31.	unpost structure	(52)
32.	unpost all structures	(92)

E. Adding or Removing PHIGS Structures

33.	open structure	(6, 14, 21, 29, 33, 41, 55, 61, 87, 93, 97)
34.	close structure	(13, 20, 28, 32, 40, 46, 58, 64, 90, 95, 99)
35.	delete structures	(49)
36.	delete structure network	(104)
37.	delete all structures	(105)

F. Adding Data to PHIGS Structure

38.	polyline 2-D	(10, 27)
39.	polyline 3-D	(11)
40.	polymarker 2-D	(36)
41.	polymarker 3-D	(39)
42.	text 2-D	(17, 18)
43.	text 3-D	(19)
44.	fill area 2-D	(31)
45.	fill area 3-D	(24)
46.	set local transform 2-D	(88)
47.	set local transform 3-D	(56, 62)
48.	set global transform 2-D	(94)
49.	set global transform 3-D	(98)
50.	execute structure	(12, 42, 43, 44, 45, 57, 63, 89)

G. Changing Characteristics of Graphic Primitives

51.	set line type	(9, 26)
52.	set line color index	(8, 25)
53.	set marker type	(35, 38)
54.	set marker color index	(34, 37)
55.	set character height	(16)
56.	set text color index	(15)
57.	set interior style	(23)
58.	set interior color index	(22, 30)

H. Transferring Data Between GSS and Applications

59.	inquire list of available workstation	(106)
60.	inquire workstation connection and type	(107)
61	inquire display space size 2-D	(109)
62.	inquire display space size 3-D	(108)

4.2 Test Methods and constraints

The tests as outlined in section 5 must be followed. The required input is stated for each test. This test procedure uses the normal operation of the functions, and does not exercise all the possible error combinations.

4.3 Test Progression

The progression of testing is fully outlined in section 5 of this unit test plan. These steps should be followed exactly to insure successful testing of the IISS configuration item.

4.4 Test Evaluation

The test results are evaluated by comparing the information returned on the various output screens to that specified as successful for the given test. As outlined in section 5, each test of the Graphic Support System functionality will provide an input screen with the required data entry specified and the resulting output for a successful test.

SECTION 5

TEST PROCEDURES

5.1   Test Description

A general description of this test was supplied in Section 3 of this manual.

5.2   Test Control

As outlined, this unit test may be done manually or run automatically using a supplied script file. To manually perform this unit test would require the tester to be logged into the IISS system and enter SDARTESTZZ on the IISS function screen.

5.3   Test Procedures

This unit test uses the ARTEST program and its associated forms. The specific forms (before and after each test step) are supplied after the listing of test steps (pages 5-6 through 5-233). They allow one to fully interrogate the 2 and 3-dimensional graphics abilities of the system.

The commands which have been made available are those routines which are described in the Form Processor Development Specification (ref. [3]). A complete description of these commands is supplied in Appendix A.

The test steps will proceed as follows:

    1)   set up graphics screens: graphics

    2)   Open PHIGS: popph

    3)   Open workstation: popwk 1 .screen.gf1.w1

    4)   set workstation window: pswkw 1 0 1 0 .50908675

    5)   set workstation viewport: pswkv 1 0 65535 0 33363

    6)   open workstation 2:   popwk 2 w2

-- make structure #1 with X-Y-Z space triad

      7)   open structure: popst 1
      8)   make lines RED: psplci RED
      9)   make solid lines: psln SOLID
     10)   add X->O->Y polyline: ppl 3 .8 .2 .2 .2 .2 .8
     11)   add O->Z polyline: ppl3 2 .2 .2 0 .2 .2 .8
     12)   display labels: pexst 2
     13)   close the structure: pclst

-- make structure #2 with labels for triad

5-1

```
        14) open structure:          popst 2
        15) make characters blue:    pstxci BLUE
        16) set height=small:        pschh .1
        17) add X text:              ptx .8 .2 X
        18) add Y label:             ptx .2 .8 Y
        19) add Z label:             ptx3 .2 .2 .8 0 0 1 0 0 1 Z
        20) close structure:         pclst
```

-- place a green triangle fill offset by Z=1.

```
        21) open structure:          popst 3
        22) fill color green:        psici GREEN
        23) set interior style:      psis SOLID
        24) fill area:               pfa3 3 0 0 1 1 0 1 0 1 1
```

-- make structure #3 as 2-D wire frame square box

```
        25) make white lines:        psplci WHITE
        26) make lines dashed:       psln DASH
        27) make box:                ppl 5 0 0 1 0 1 1 0 1 0 0
        28) close structure:         pclst
```

-- make structure #4 as yellow filled diamond inside #3

```
        29) open structure:          popst 4
        30) fill color yellow:       psici YELLOW
        31) add region:              pfa 4 .5 0 1 .5 .5 1 0 .5
        32) close structure:         pclst
```

-- make structure #5 as red stars at vertices of #4

```
        33) open:                    popst 5
        34) marker color red:        pspmci RED
        35) marker type to star:     psmk STAR
        36) position:                ppm 4 .5 0 1 .5 .5 1 0 .5
```

-- place MAGENTA circles at vertices of fill in #3

```
        37) marker color magenta:    pspmci MAGENTA
        38) marker type=circle:      psmk CIRCLE
        39) 3-D positions:           ppm3 3 0 0 1 1 0 1 0 1 1
        40) close structure          pclst
```

-- make structure displaying above definitions

```
        41) open:                    popst 6
        42) yellow fill in box:      pexst 4
        43) box w/ fill area:        pexst 3
        44) markers in structures:   pexst 5
        45) display triad:           pexst 1
        46) close structure          pclst
```

```
47) display from 2-D view:      ppost 1 6 1
48) post in smaller window:     ppost 2 6 1
```

```
49) delete labels from triad: pdst 2
```

```
50) update display in #1:    puwk 1 PERFORM
51) update display on #2:    puwk 2 PERFORM

52) remove graphic display:  pupost 1 6
53) update empty workstation: puwk 1 PERFORM
```

-- rotate picture 45 degrees about X axis

```
54) make rotation matrix:    prox 45 A3
55) open new structure:      popst 7
56) add local matrix:        pslmt3 A3 POST
57) bring in displays:       pexst 6
58) close                    pclst
59) display rotated figure:  ppost 1 7 1
```

-- overlay this with rotate 45 degrees about Y axis

```
60) make rotation matrix:    proy 45 B3
61) make new structure:      popst 8
62) add local matrix:        pslmt3 B3 POST
63) bring in displays:       pexst 7
64) close                    pclst
65) display rotated figure:  ppost 1 8 1
```

-- test 2 dimensional transformation utilities

```
66) translate 2-D:       ptr 10 10 C2
67) rotate in 2-D:       pro 180 B2
68) compose 2-D:         pcom C2 B2 A2
69) scale 2-D:           psc 10 100 B2
70) add to matrix:       pcom A2 B2 C2
71) transform point 2D:  ptp 1 0 C2
72) transform point 2D:  ptp 0 1 C2

73) trans. matrix 2D: pbltm 10 20 -10 -20 2 4 180 A2
74) mult w/ previous: pcotm A2 -10 -20 10 20
                                    .5 .25 180 B2
75) verify back to identity: ptp 1 1 B2
```

-- test 3-D transformation utilities

```
76) translate:          ptr3 10 10 10 C3
77) rotate about Z:     proz 45 B3
78) compose 3D:         pcom3 C3 B3 A3
79) scale 3-D:          psc3 10 100 1000 B3
80) add to matrix:      pcom3 A3 B3 C3
81) transform point:    ptp3 1 1 1 C3

82) make transf. matrix: pbltm3  10 20 30  1 2 3  2 4 8
                                    180 0 0  A3
83) compose transform:   pcotm3 A3 -10 -20 -30 -1 -2 -3
                                    -180 0 0  .5 .25 .125 B3
84) verify identity:    ptp3 1 1 1 B3
```

-- check misc. items

85) set workstation window: pswkw3 1 0 1 0 .69999237 0 1

86) set workstation viewport: pswkv3 1   0 65535   0 33363
                                                    0 65535

-- check usage of 2-D local/global matrices

```
87)  open new structure #9:      popst 9
88)  set local 2D transform:     pslmt A2 POST
89)  reference other:            pexst 8
90)  close structure:            pclst
91)  display w/ transform:       ppost 1 9 1
92)  unpost all structures:      pupast 1 9

93)  append global transform:    popst 9
94)  set global 3-D matrix:      psgmt B2
95)  close structure:            pclst
96)  display (no change):        ppost 1 9 1

97)   append global transform:   popst 9
98)   identity 3-D:              psgmt3 B3
99)   close structure:           pclst
100)  display (same display):    ppost 1 9 1
```

101) redraw structures: prst 1 ALWAYS

102) update workstation: puwk 1 PERFORM

103) send message: pmsg "GSS UTP output is completed"

104) delete structure network: pdsn 6 DELETE

105) delete all structures: pdas

106) inquire available workstations: pqewk

107) inquire workstation conn. and type: pqwkc 1

108) inquire display space size 3-D: pqdsp3 w1

109) inquire display space size 2-D: pqdsp w1

110) error handler: perhnd 0 0
110a) examine message queue results

111) error log: perlog 0 0
111a) examine message queue results

112) close workstation 1:   pclwk 1

113) close workstation 2:   pclwk 2

111) close PHIGS: pclph
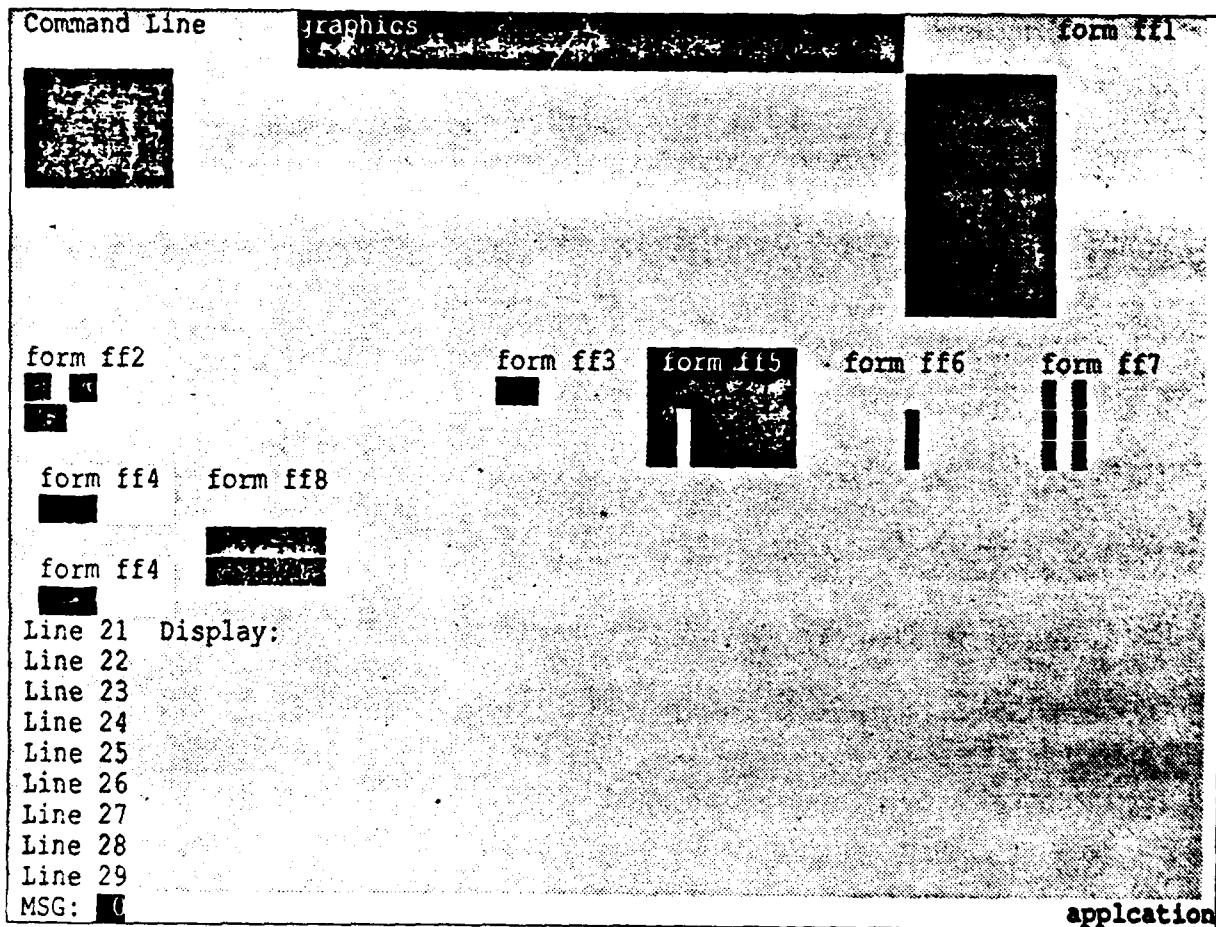
Figure 5-1a   (BEFORE)

Command Line   form gfl

Step
Output

MSG: 0   applcation

Figure 5-1b   (AFTER)

Command Line ... bopph ... form gfl

Step
Output

MSG: C applcation
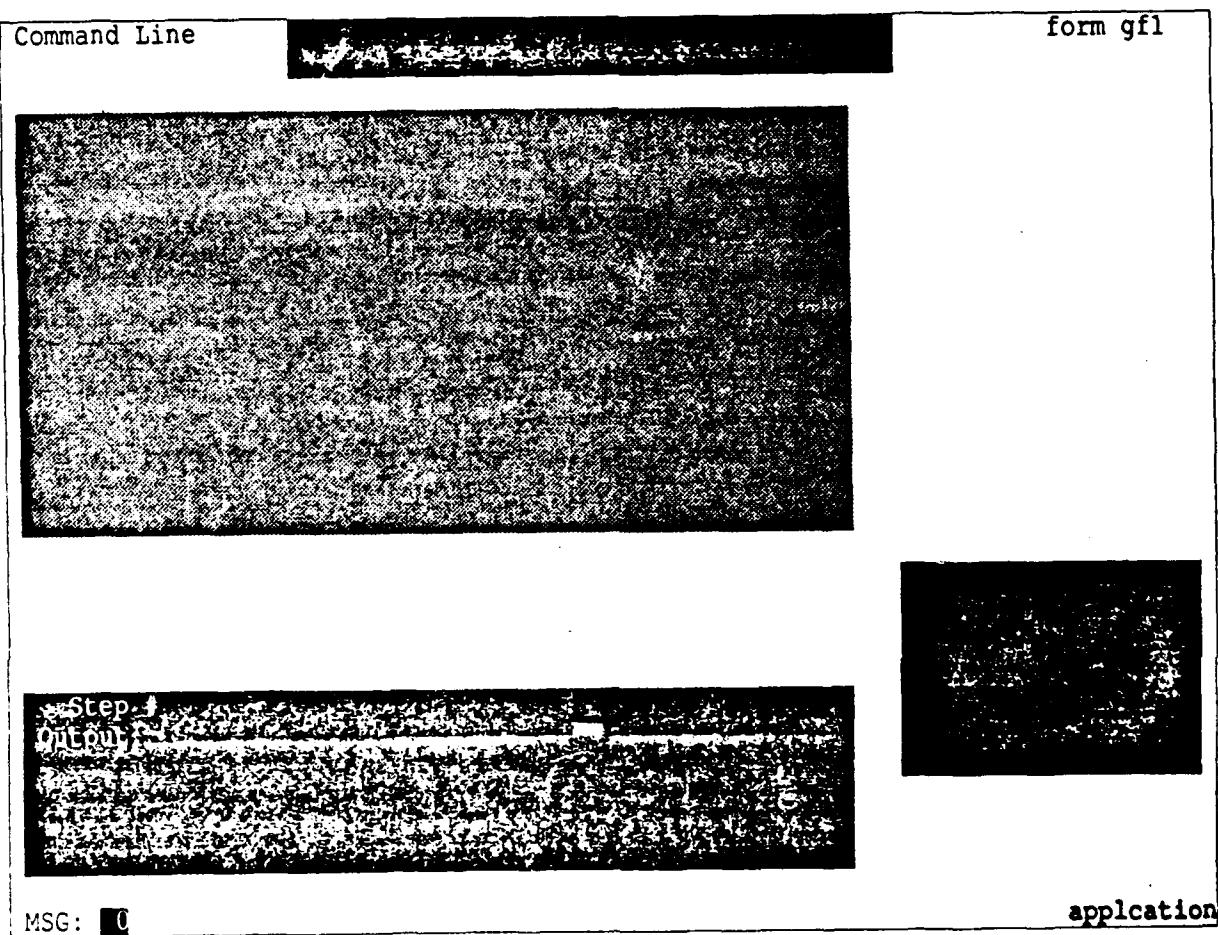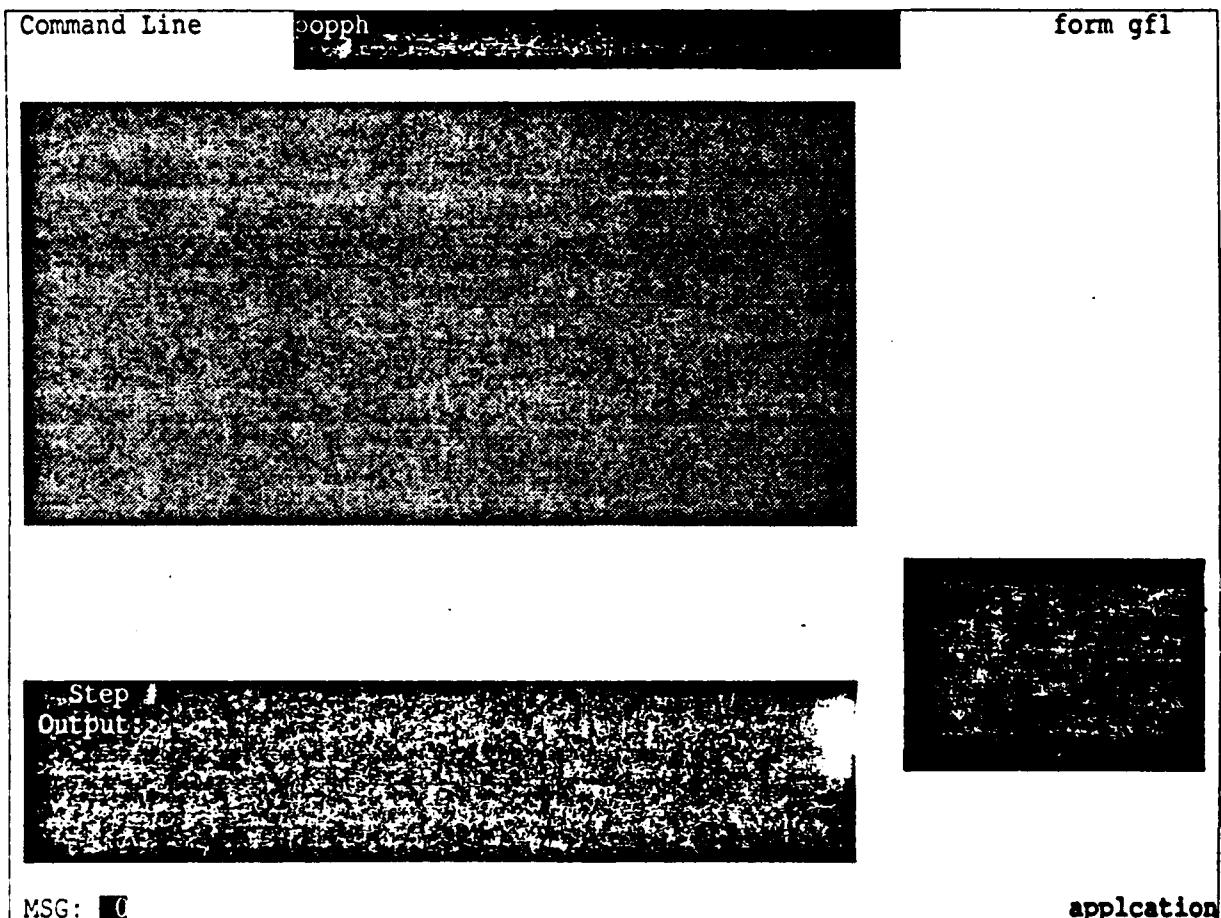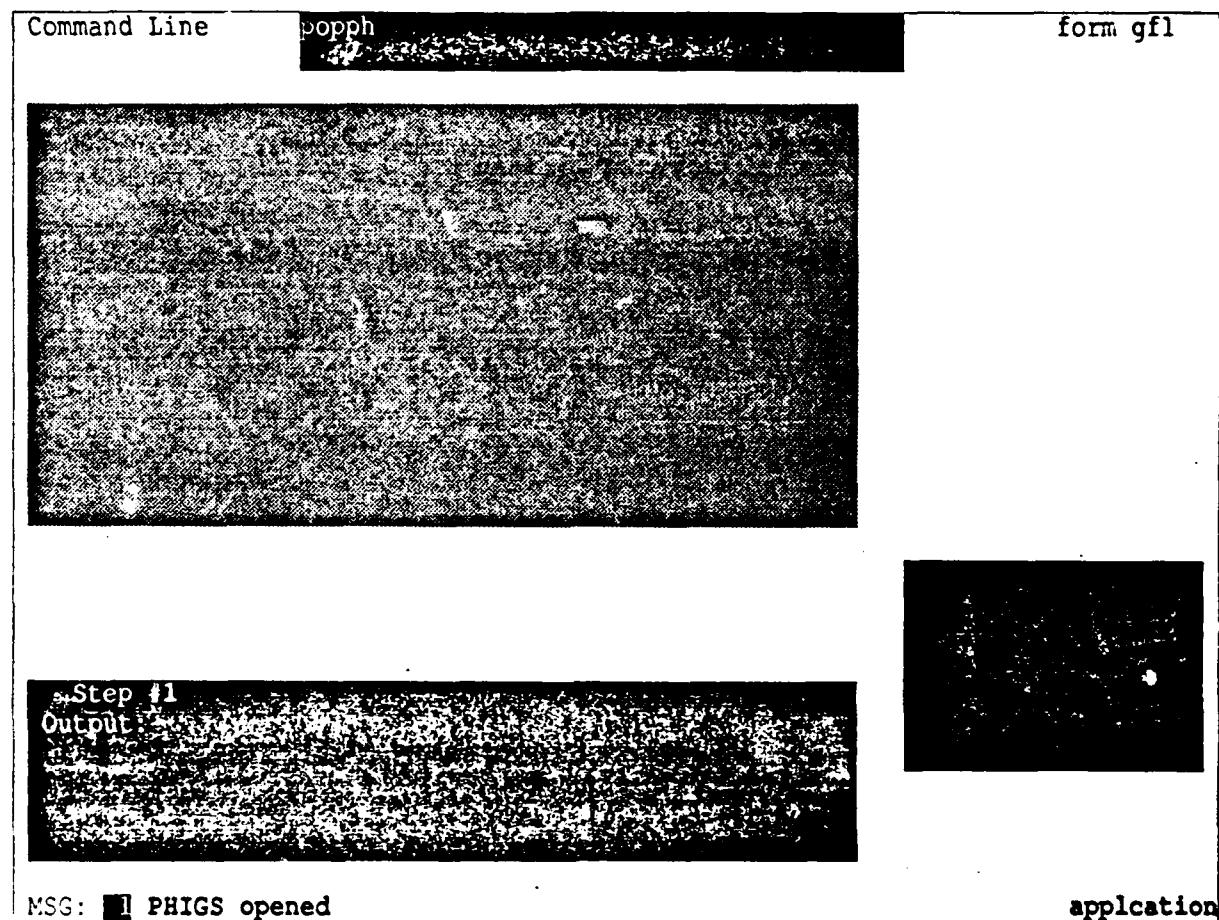
Figure 5-2a   (BEFORE)

Figure 5-2b   (AFTER)
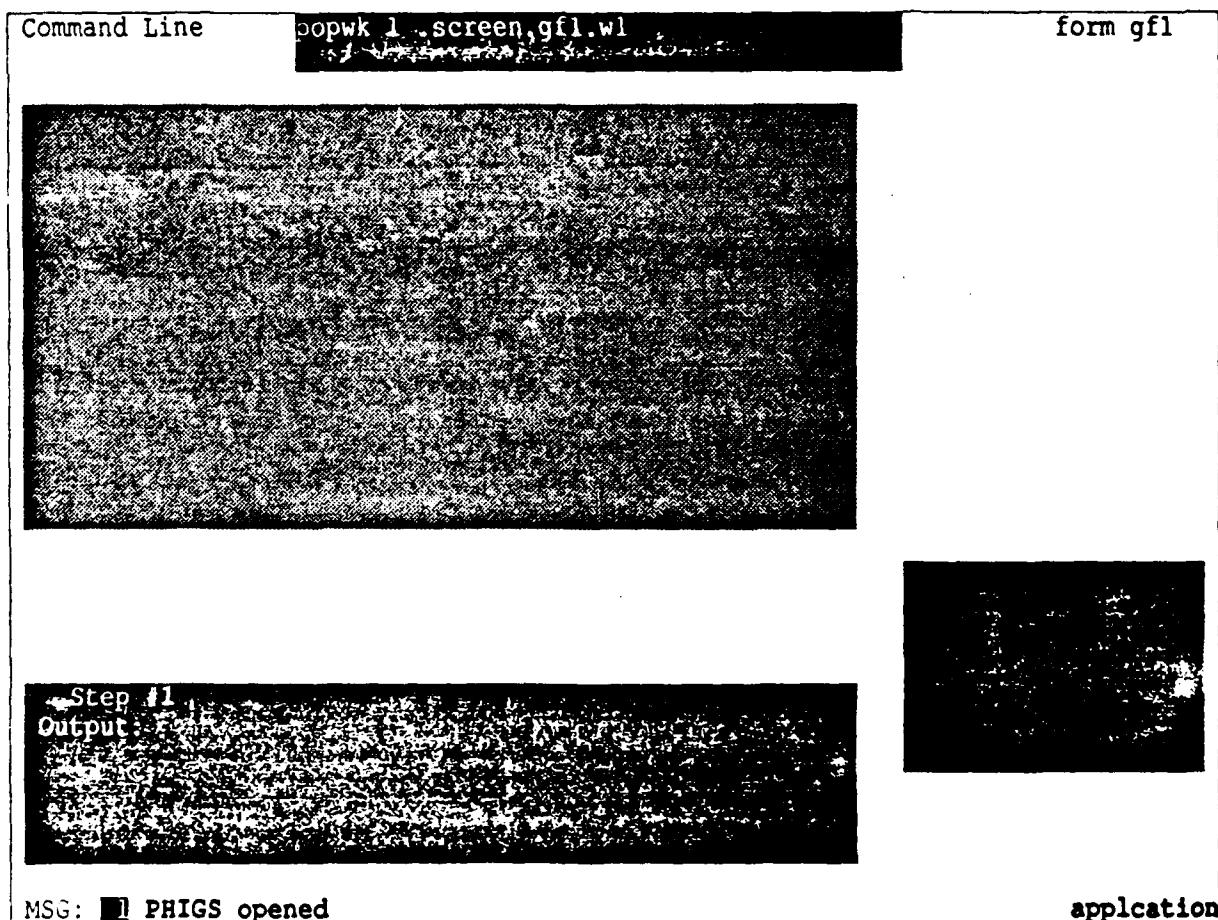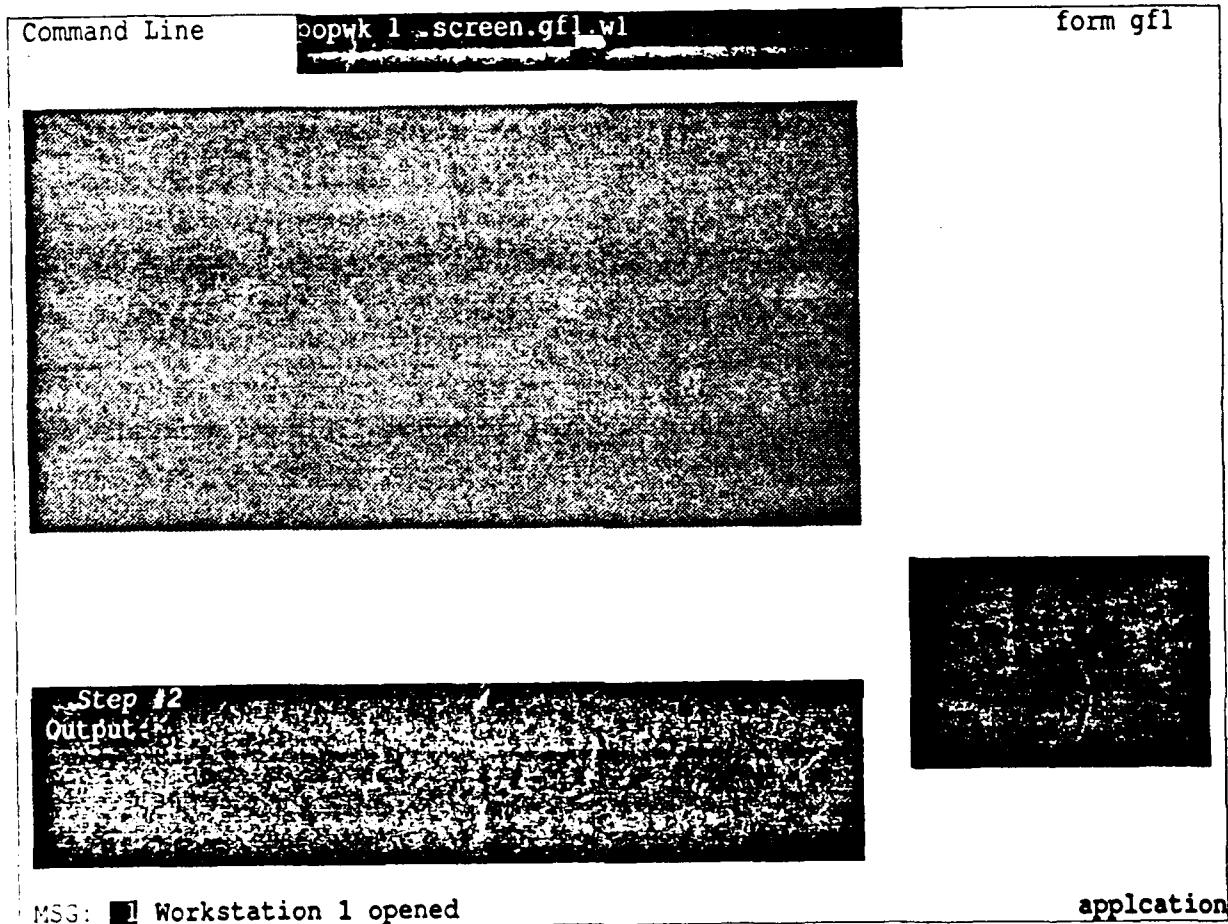
Figure 5-3a   (BEFORE)
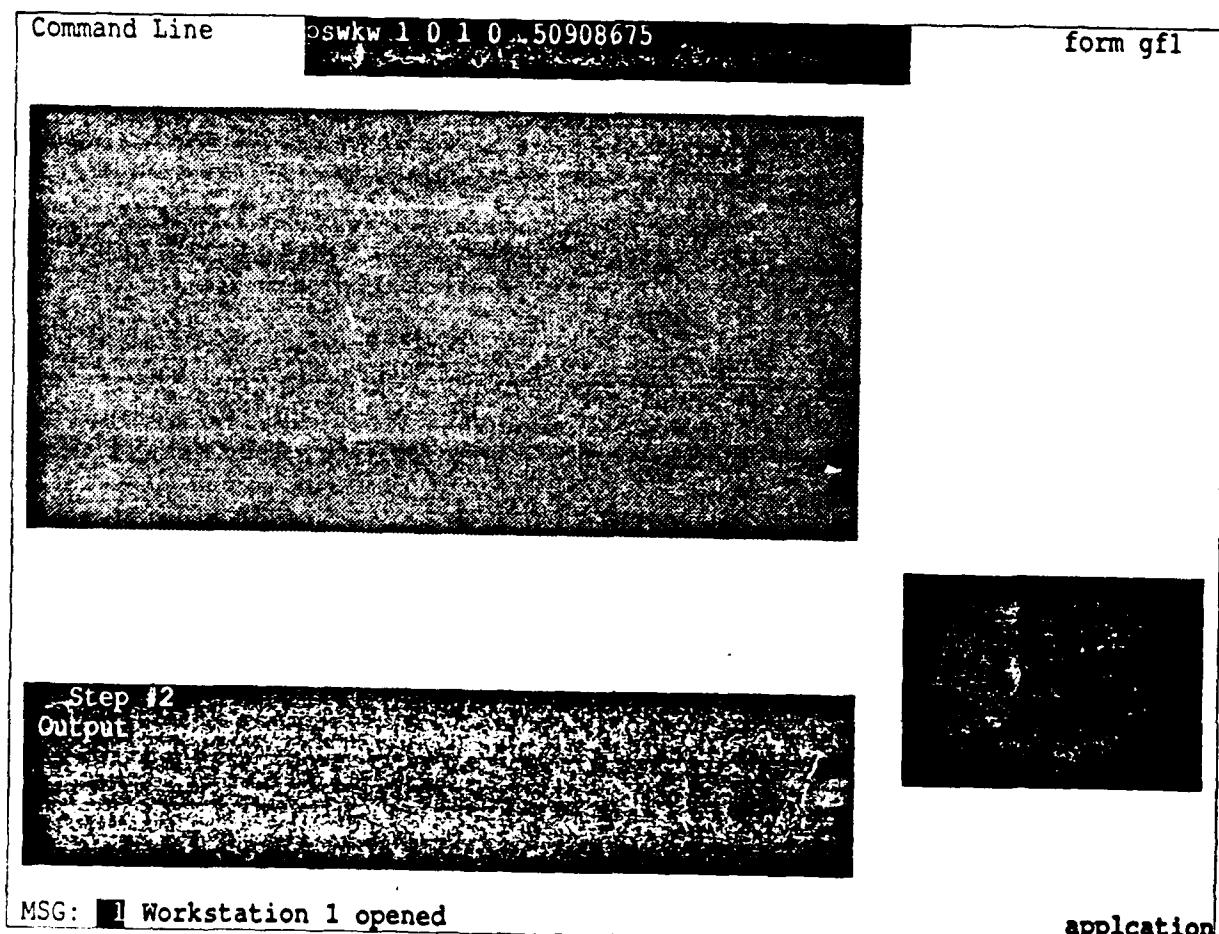
Figure 5-3b   (AFTER)
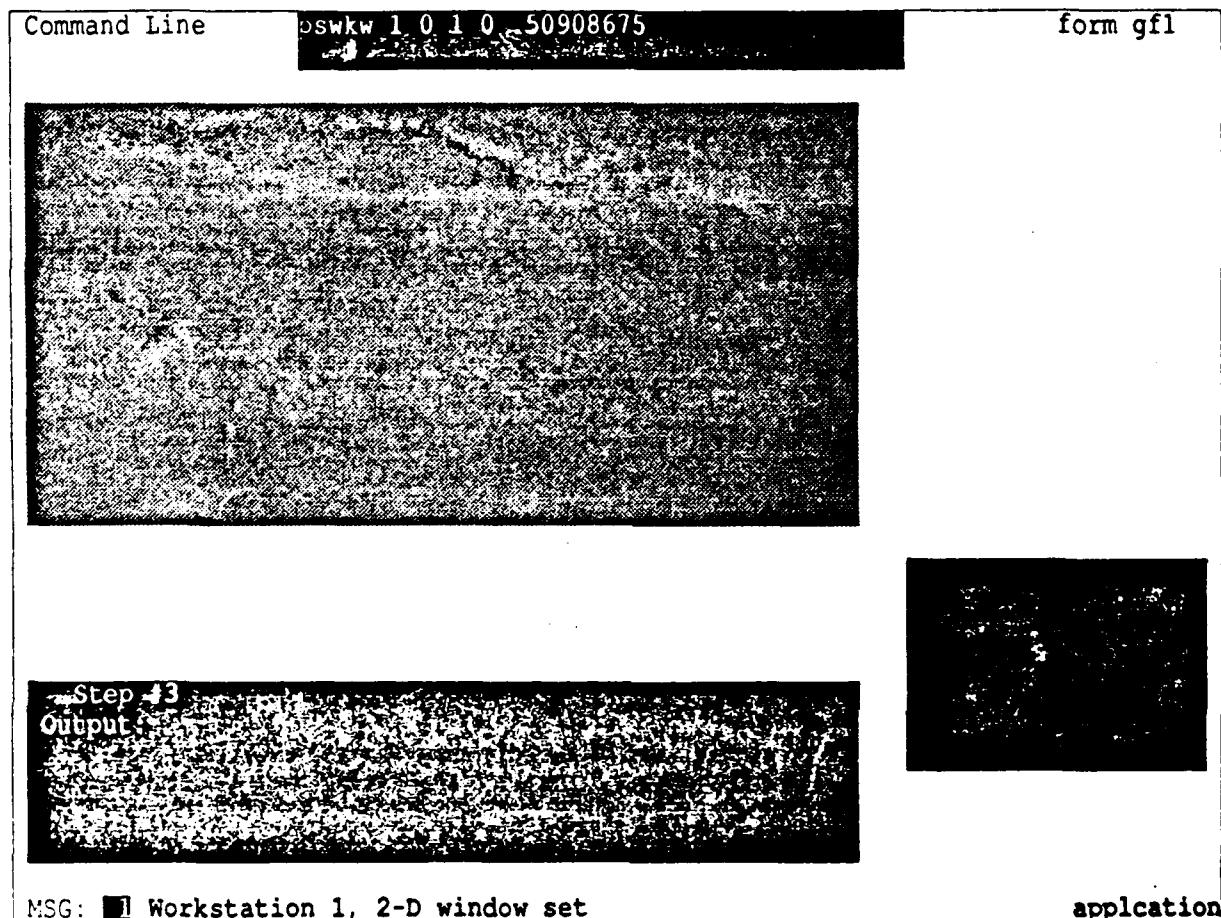
Figure 5-4a   (BEFORE)

Figure 5-4b   (AFTER)

Figure 5-5a   (BEFORE)

Command Line   pswkv 1 0 65535 0 33363   form gfl

Step #4
Output

MSG: ▋ Workstation 1, 2-D viewport set   applcation
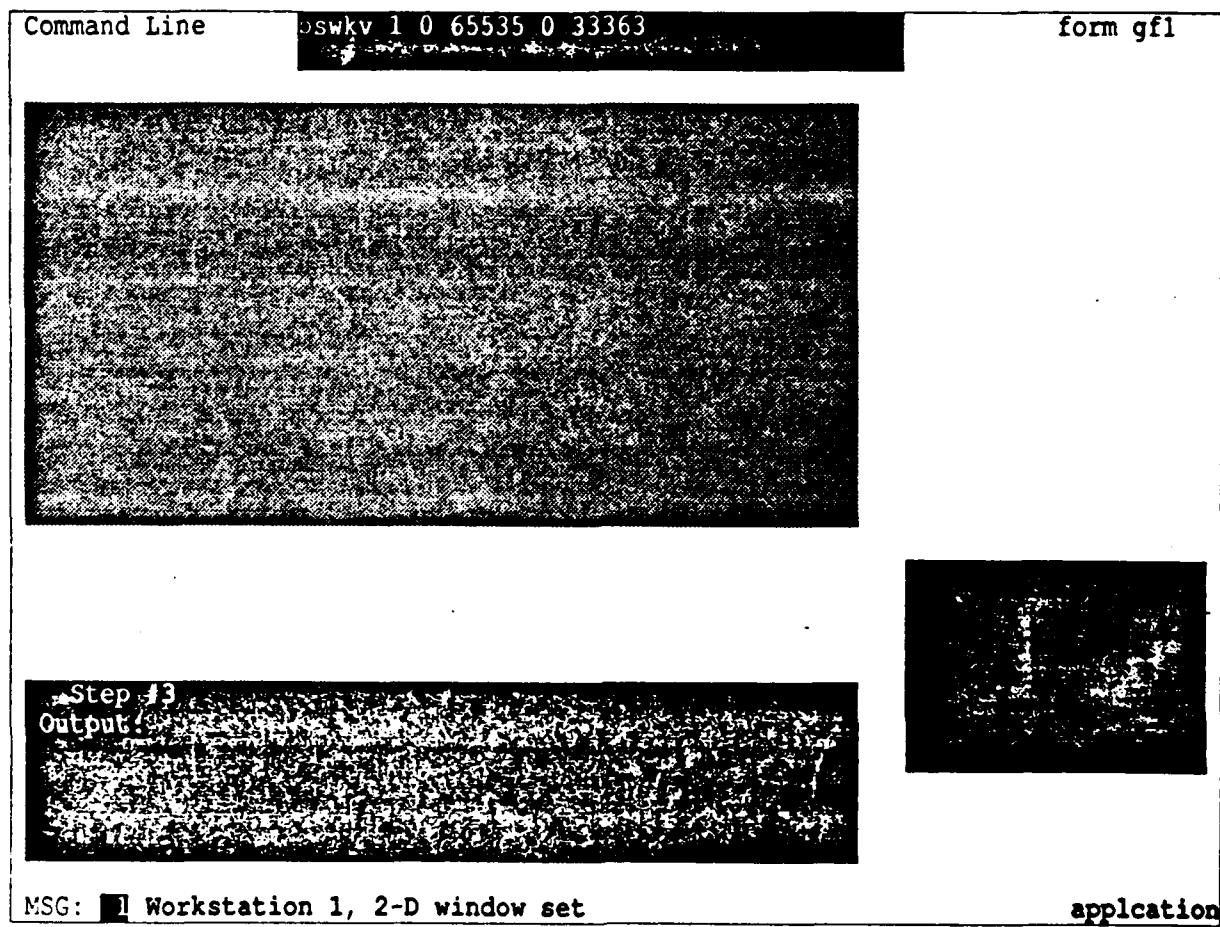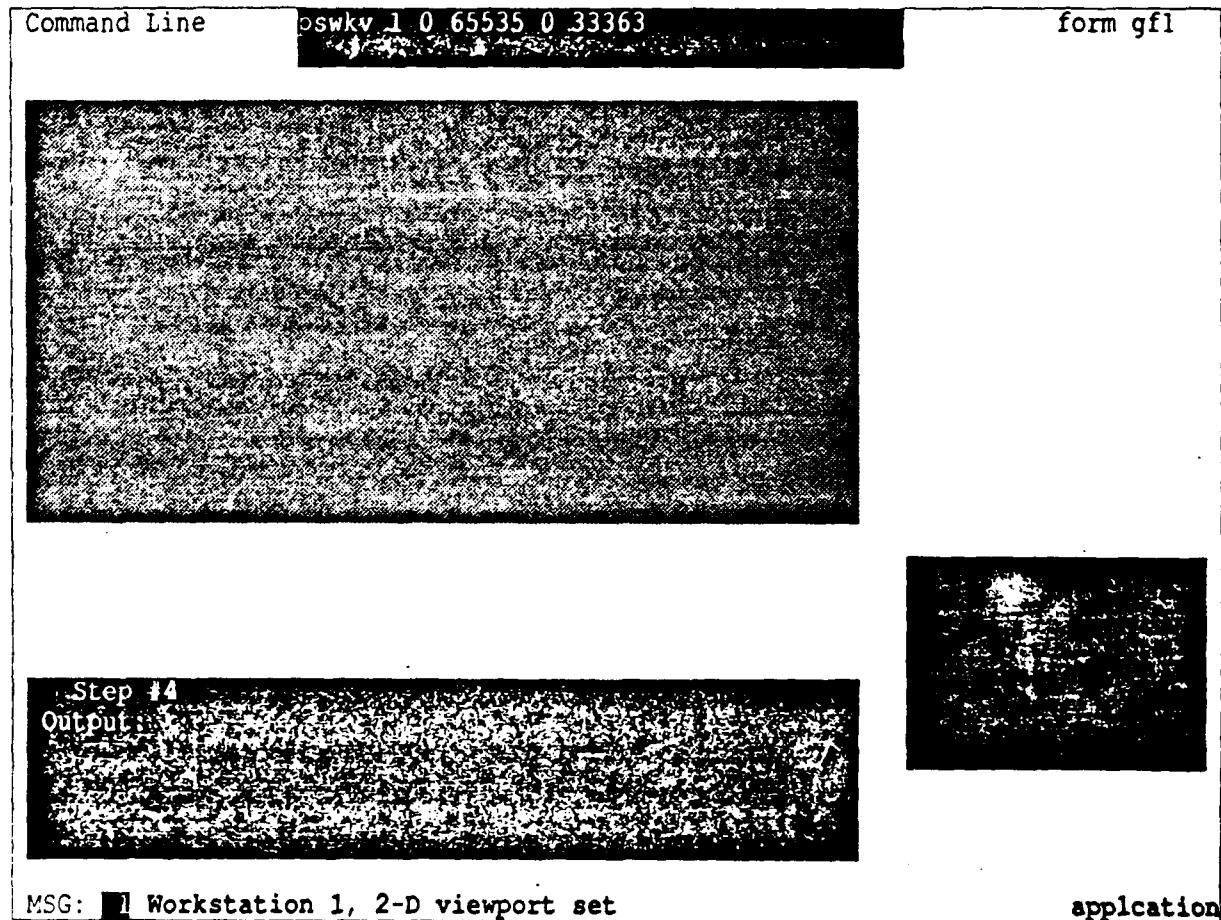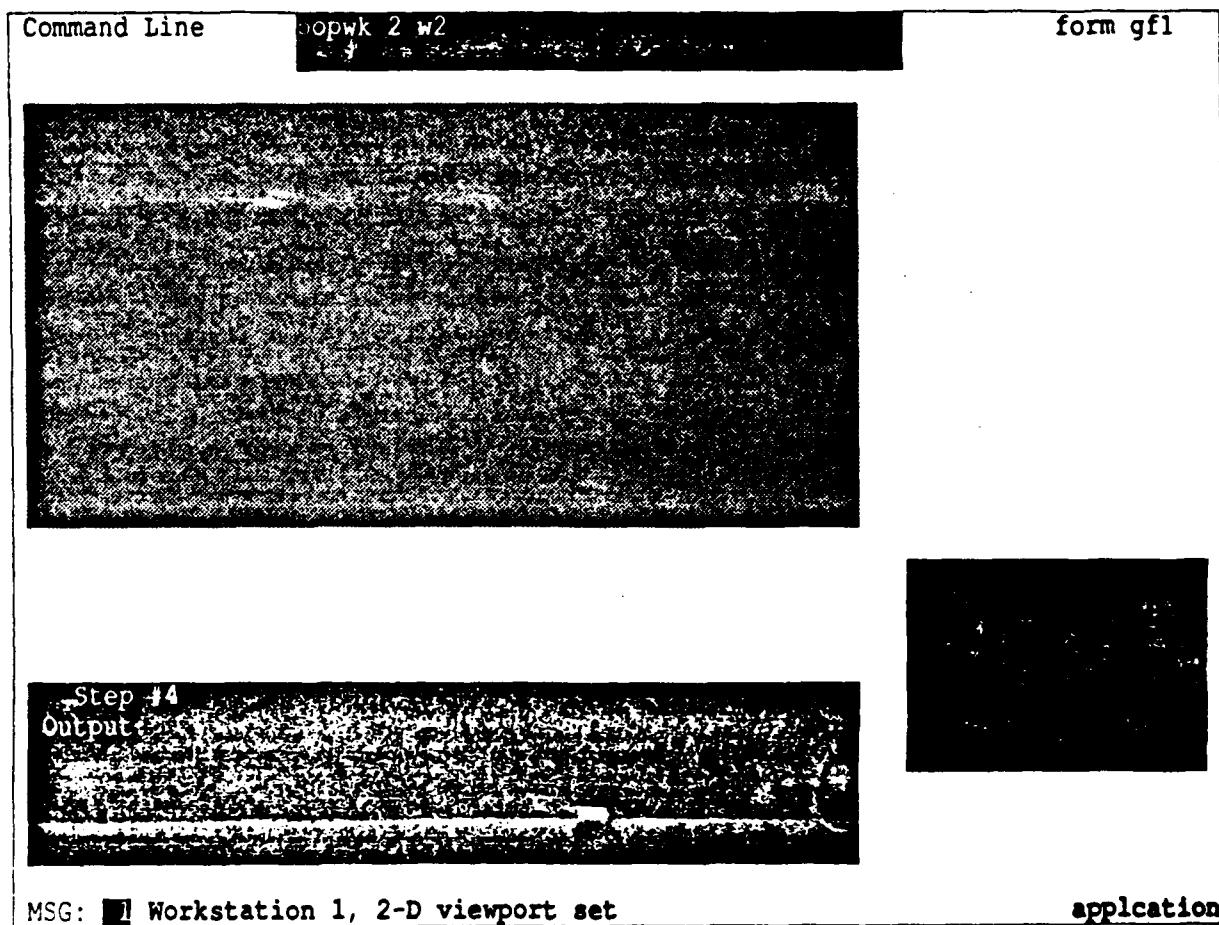
Figure 5-5b   (AFTER)

Figure 5-6a   (BEFORE)

Figure 5-6b   (AFTER)

Command Line   opst 1   form gfl

Step 5
Output

MSG: Workstation 2 opened   applcation
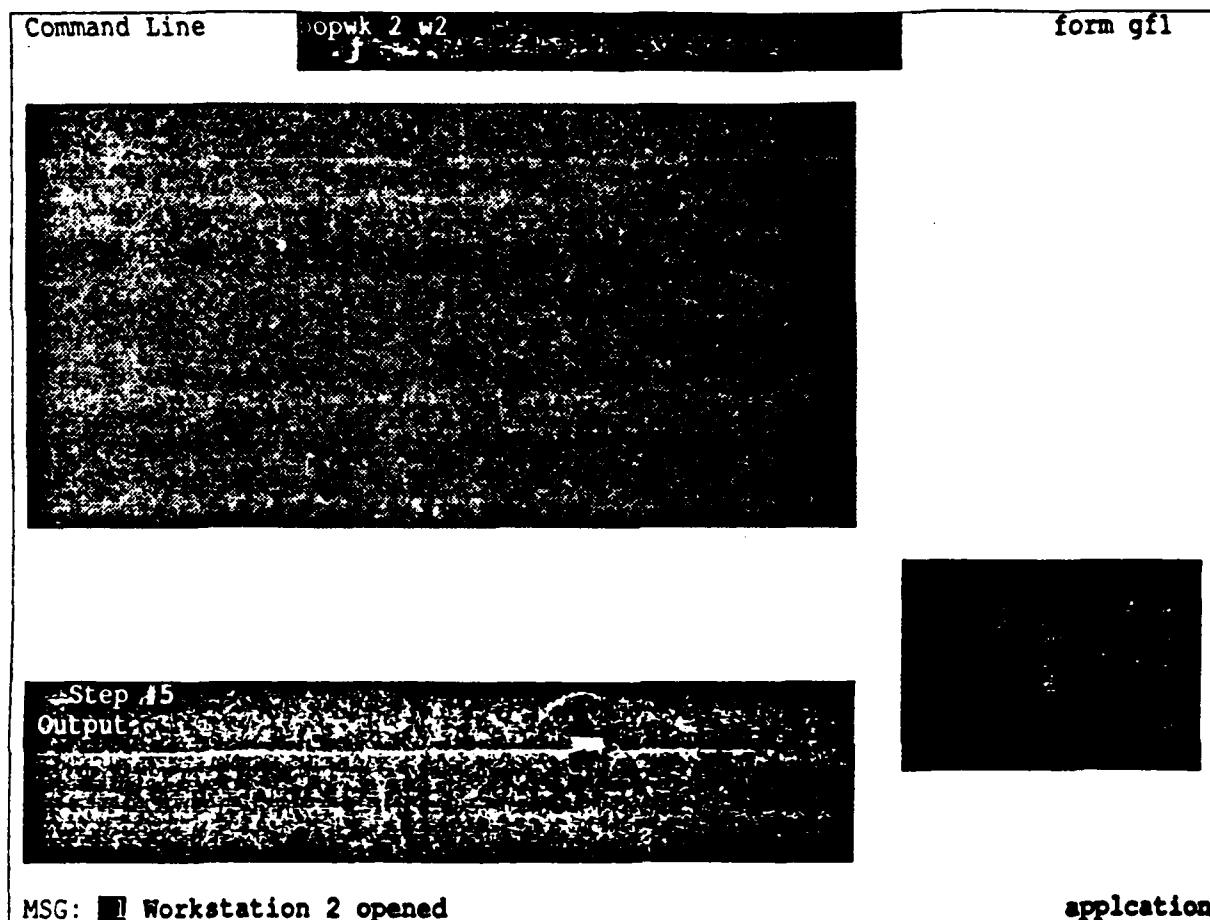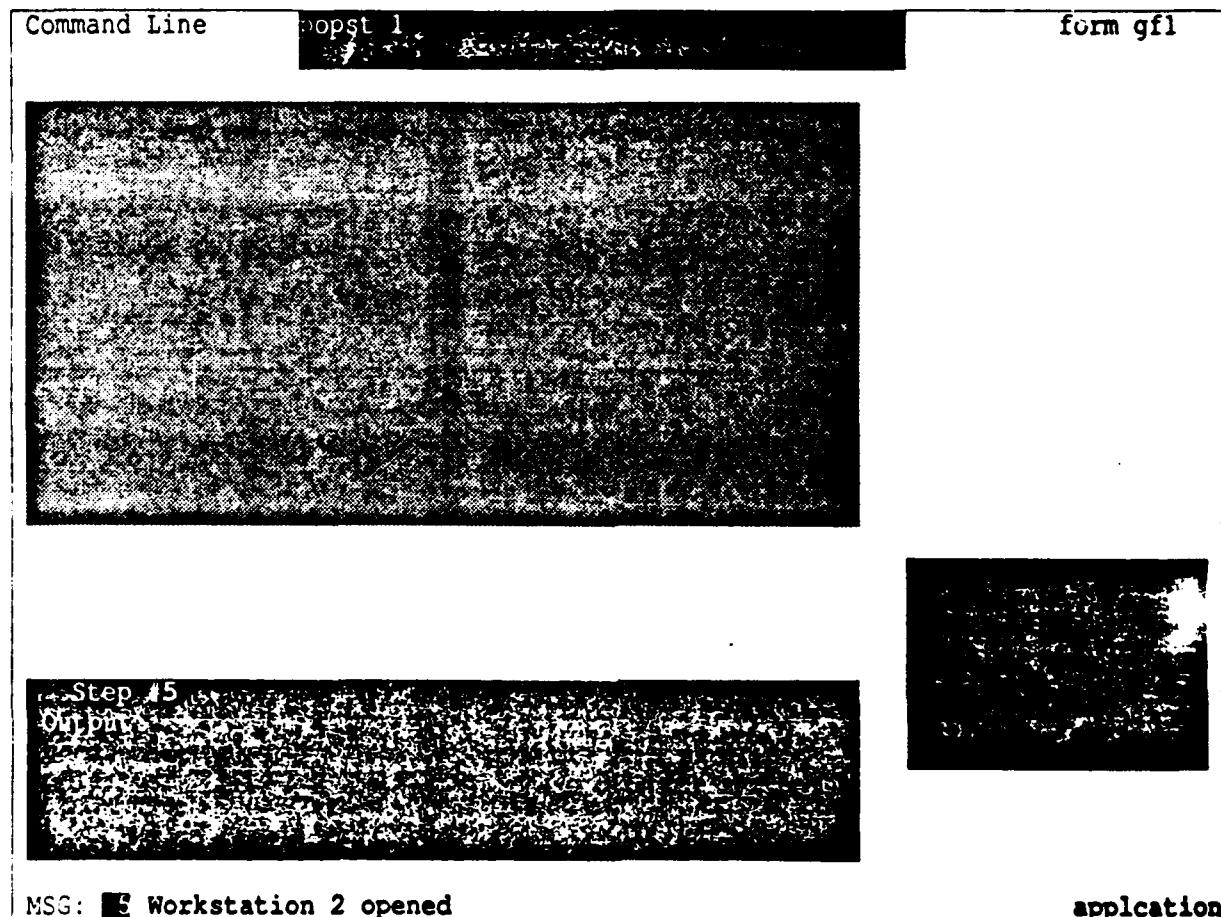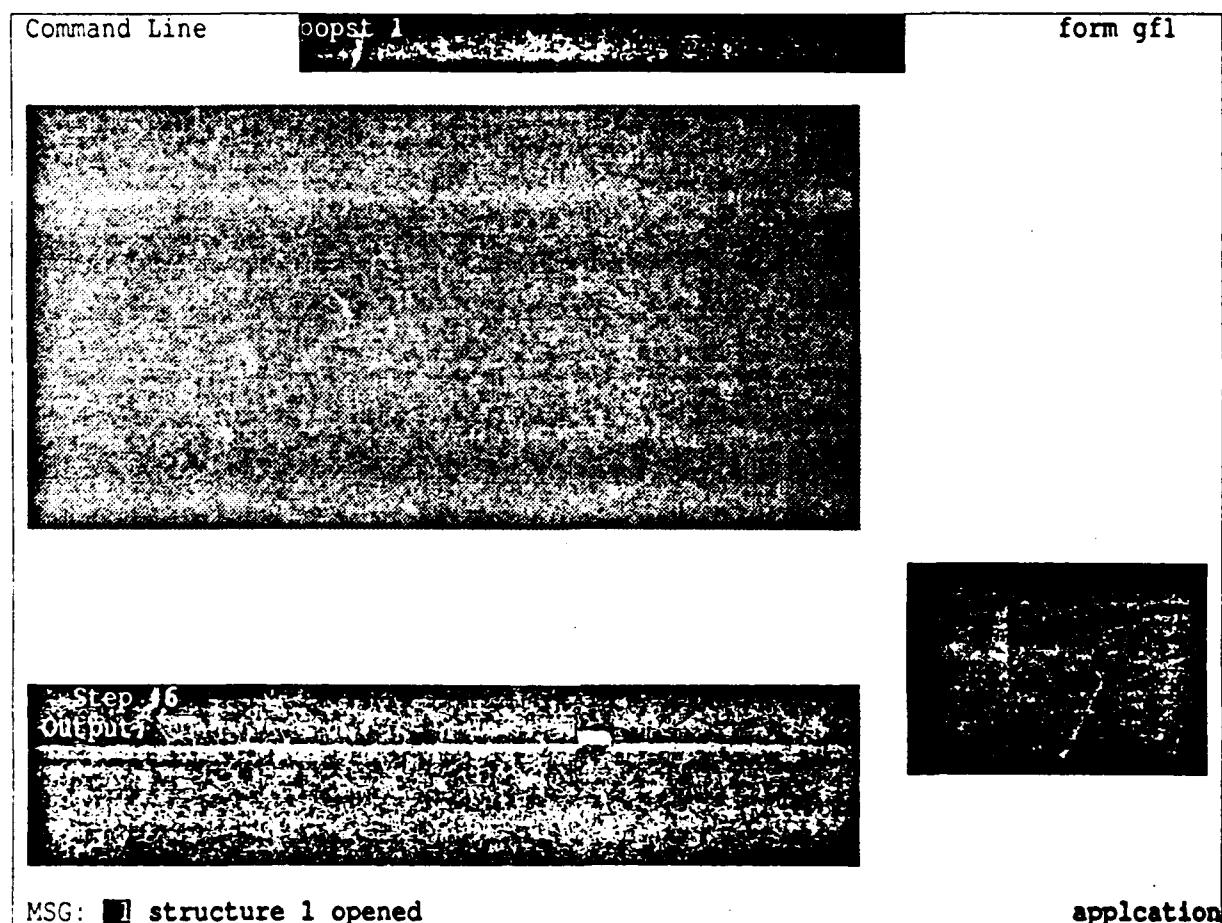
Figure 5-7a   (BEFORE)

Figure 5-7b   (AFTER)
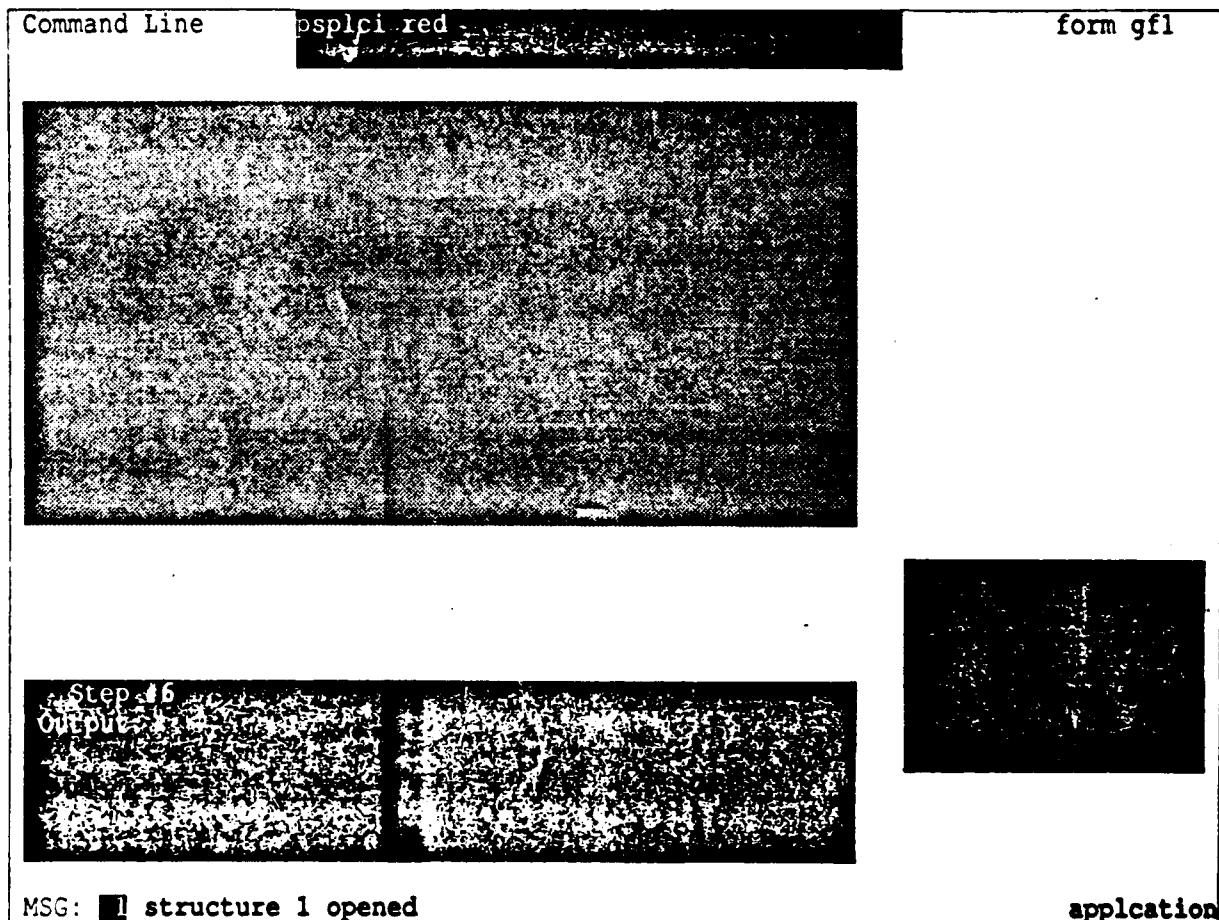
Figure 5-8a   (BEFORE)
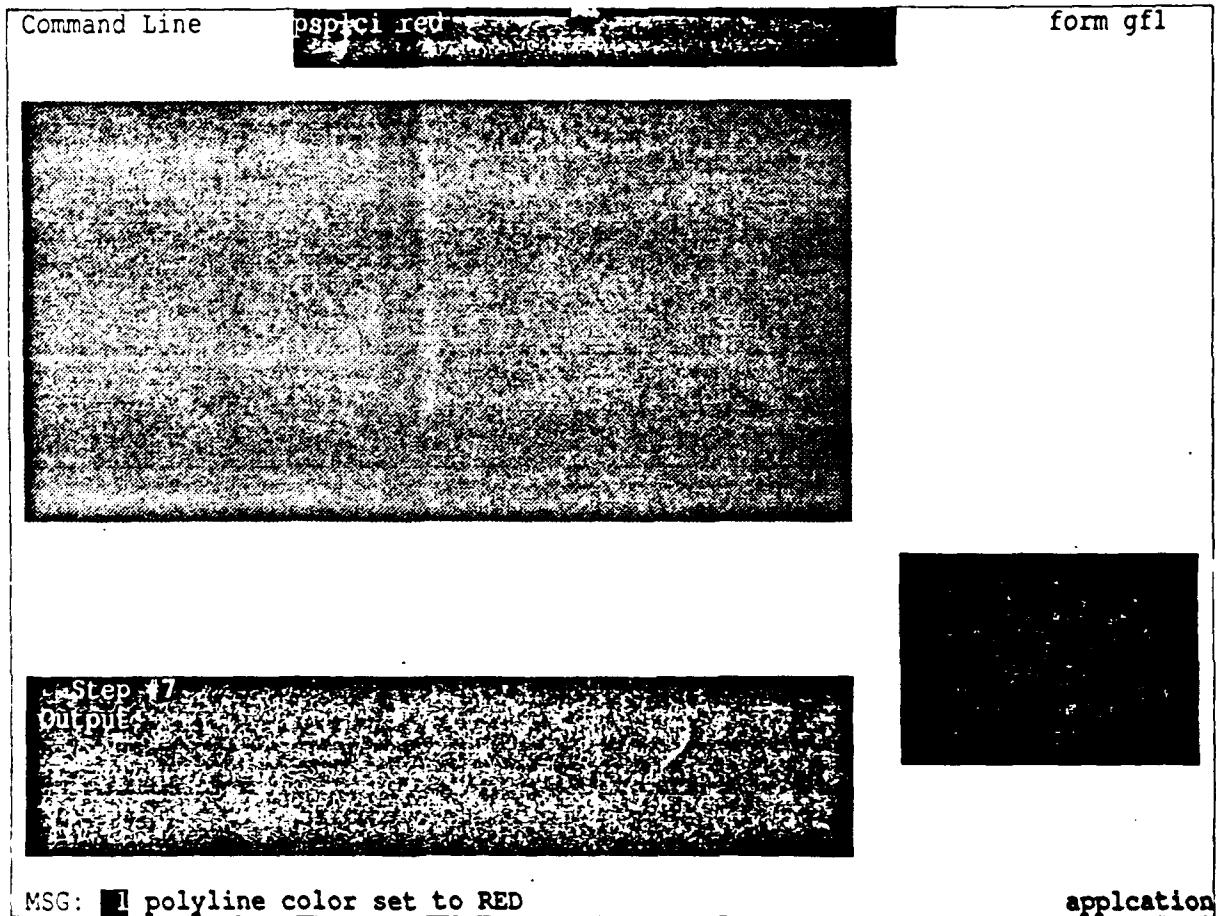
Figure 5-8b   (AFTER)

Figure 5-9a   (BEFORE)

Command Line   psln solid   form gfl

Step #8
Output

MSG: █l polyline style set to SOLID   **applcation**
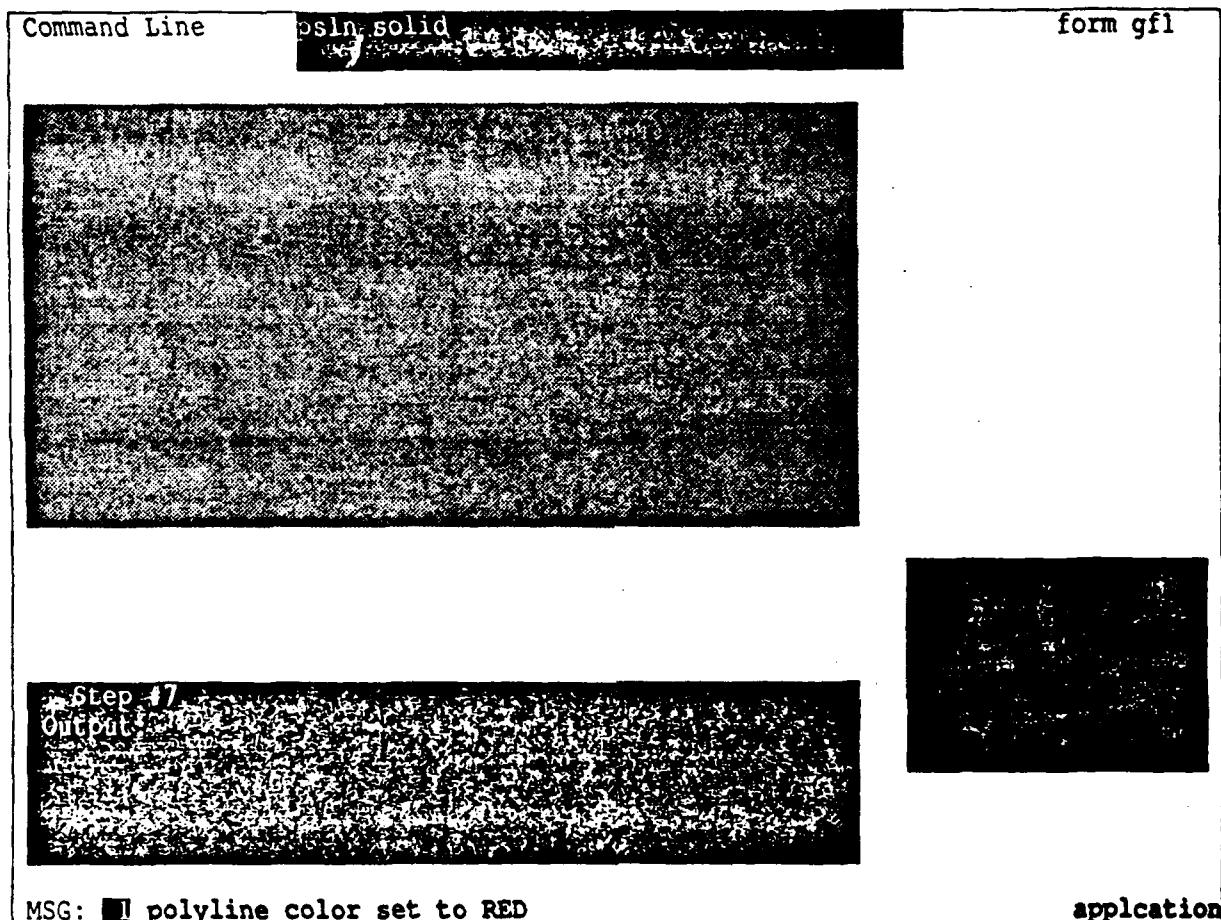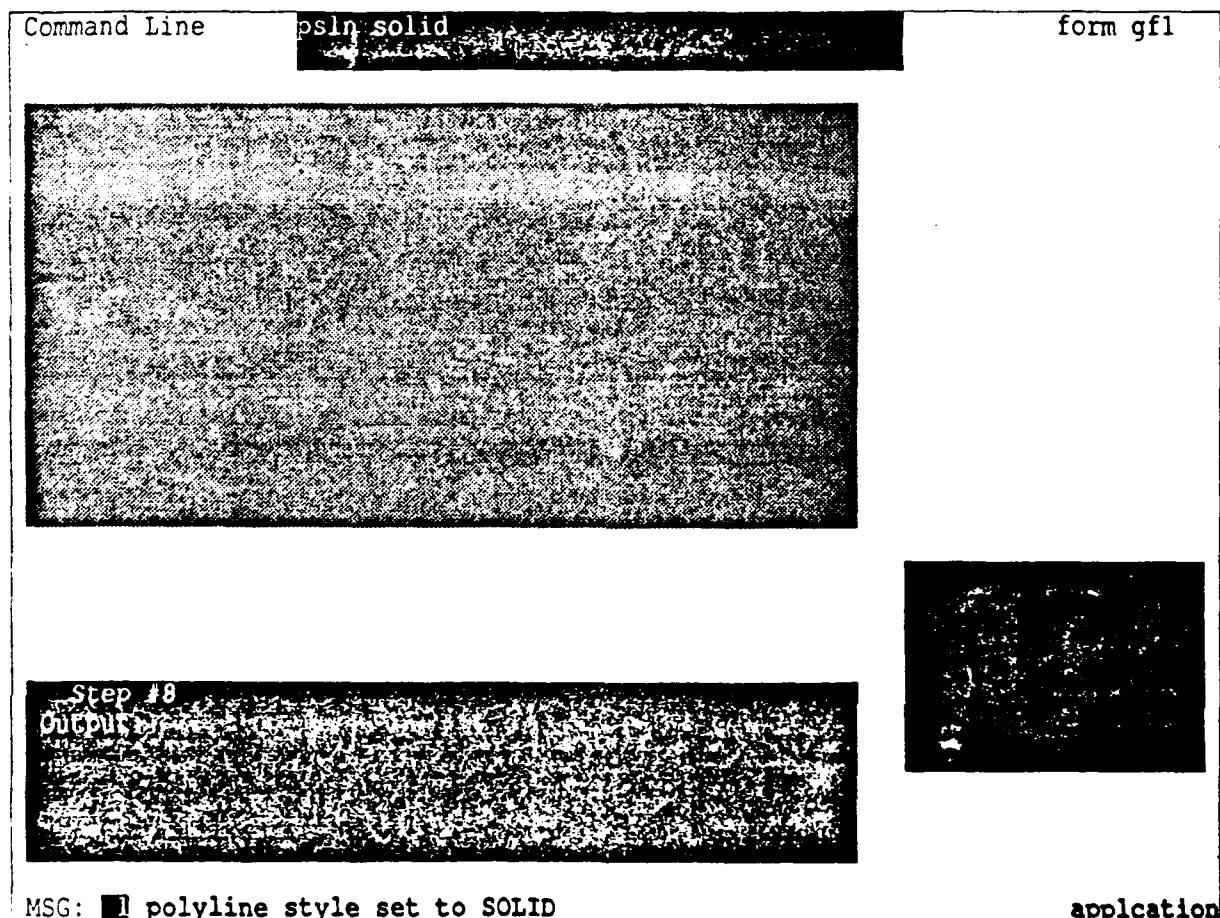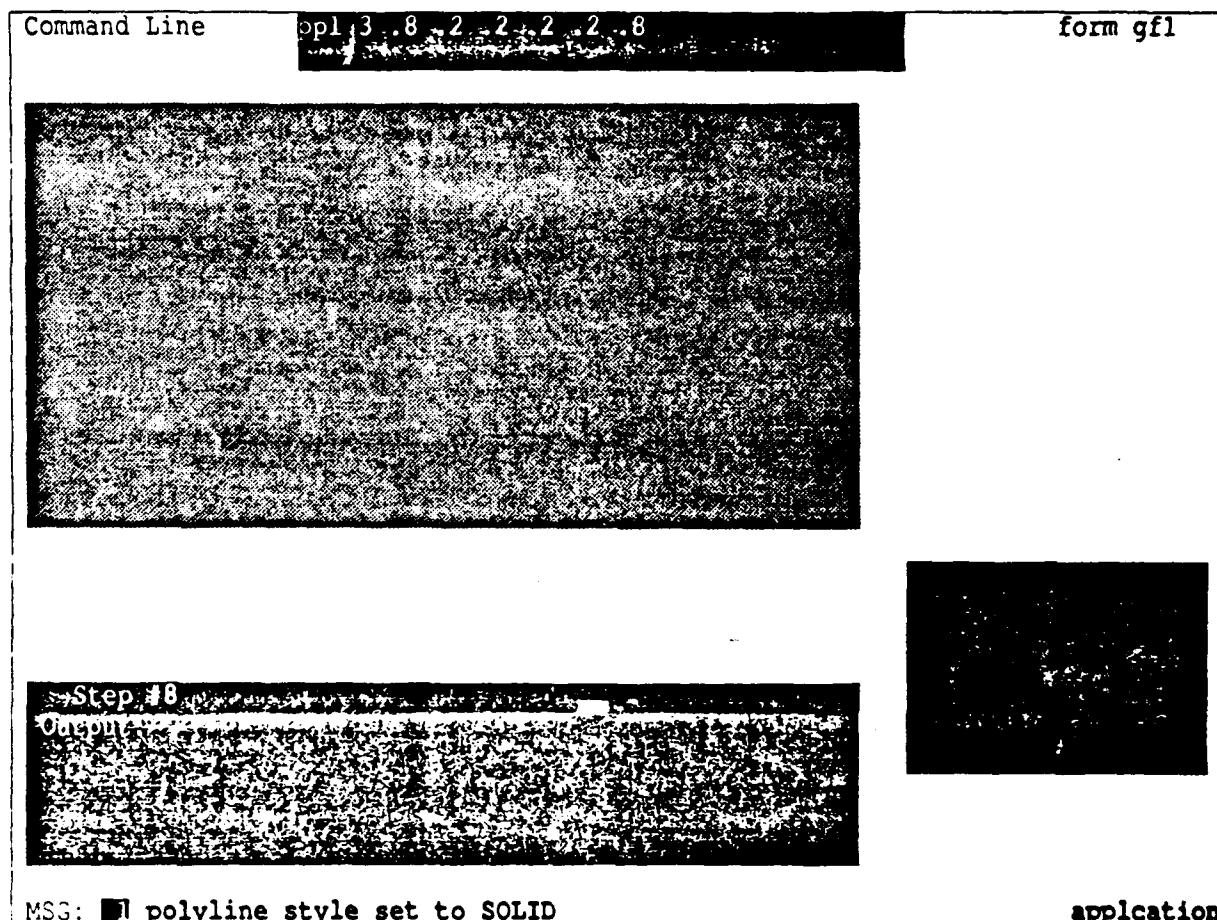
Figure 5-9b   (AFTER)

Figure 5-10a   (BEFORE)
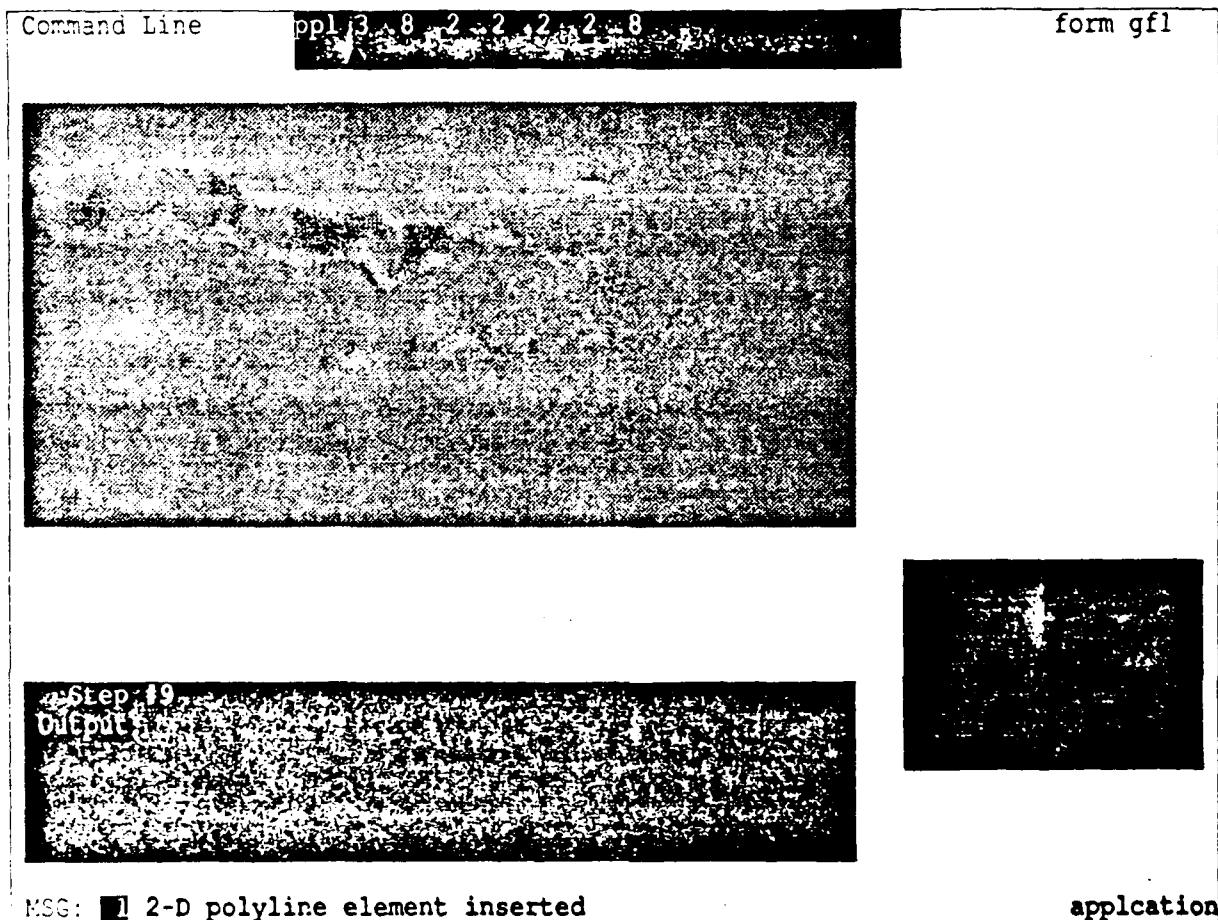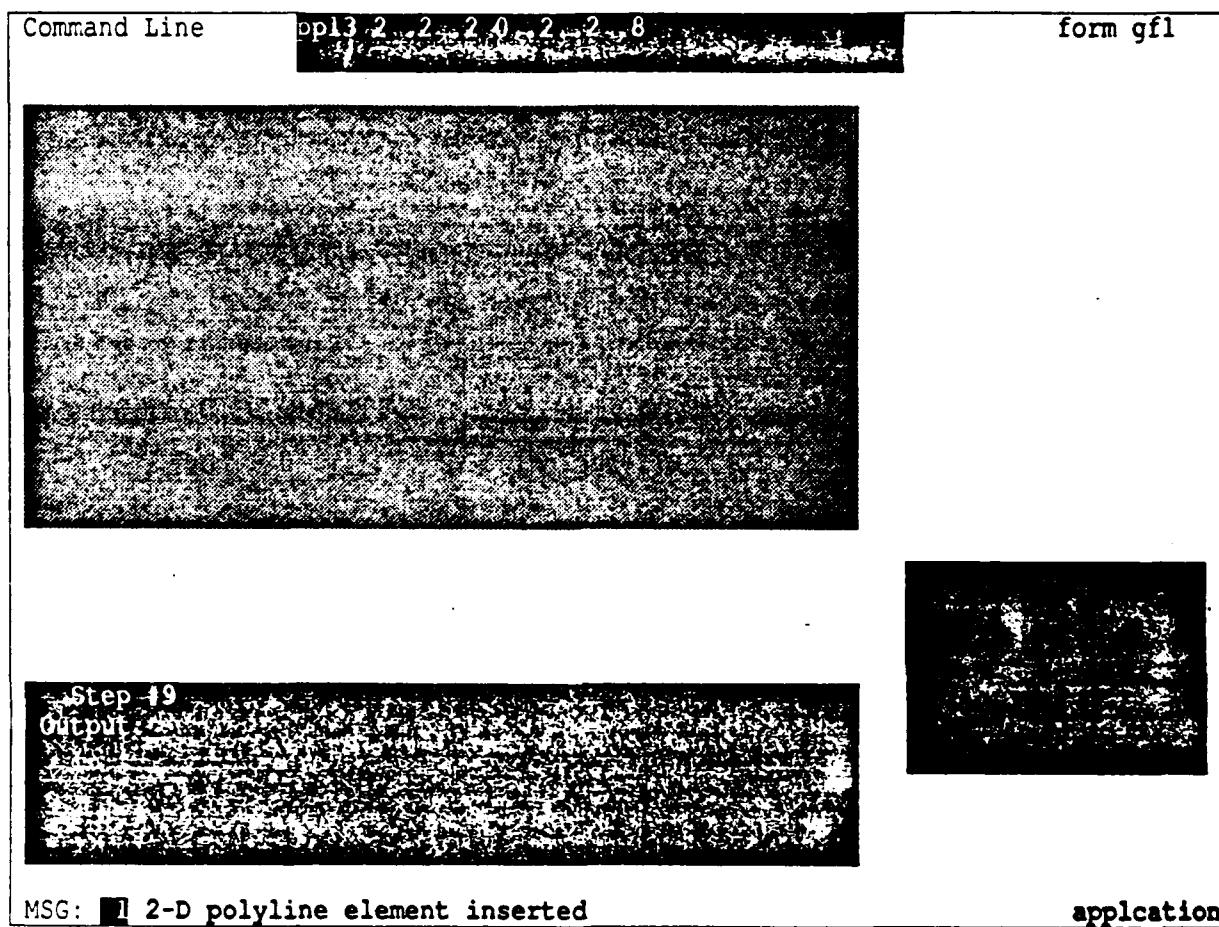
Figure 5-10b   (AFTER)

Figure 5-11a   (BEFORE)

Figure 5-11b   (AFTER)

Command Line   pexst 2   form gfl

Step #10
output

MSG: ■ 3-D polyline element inserted   applcation
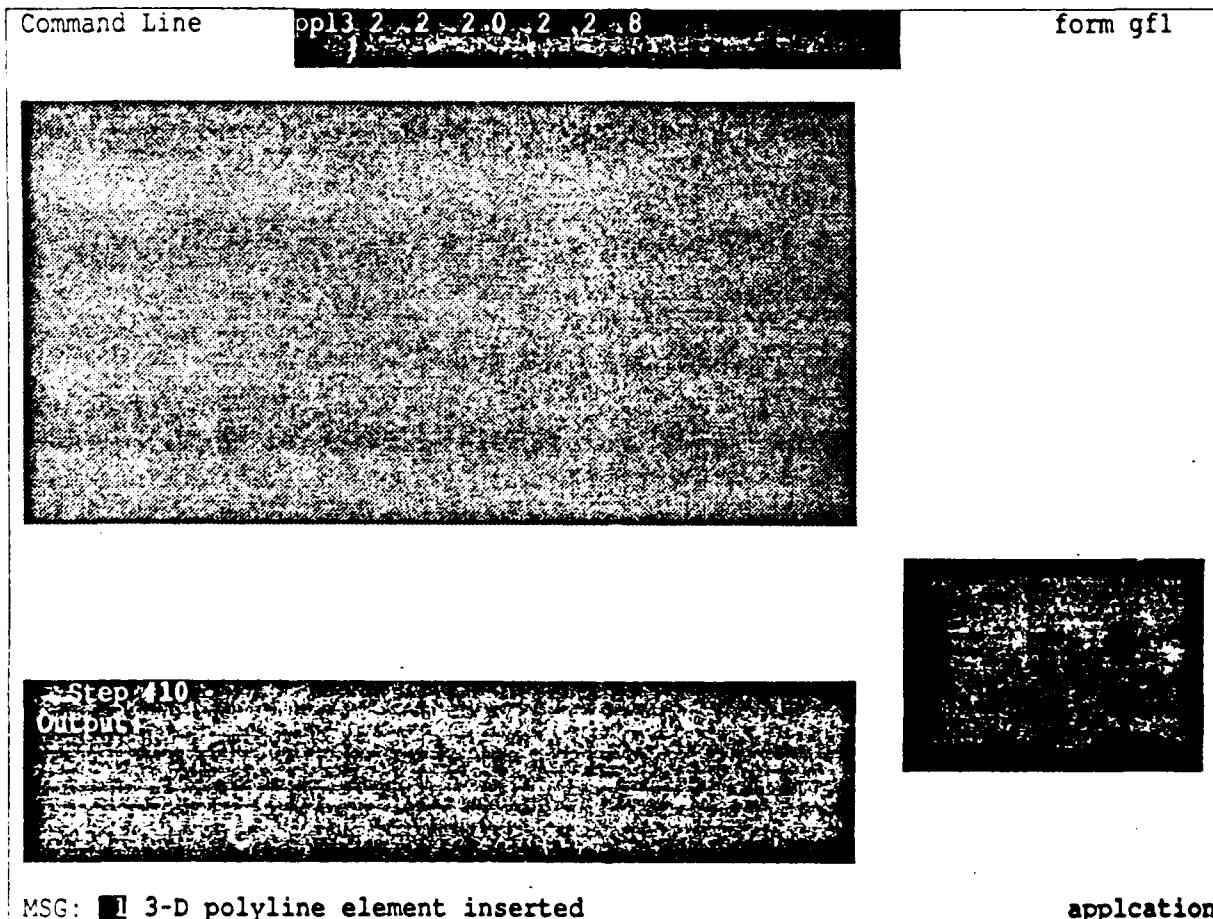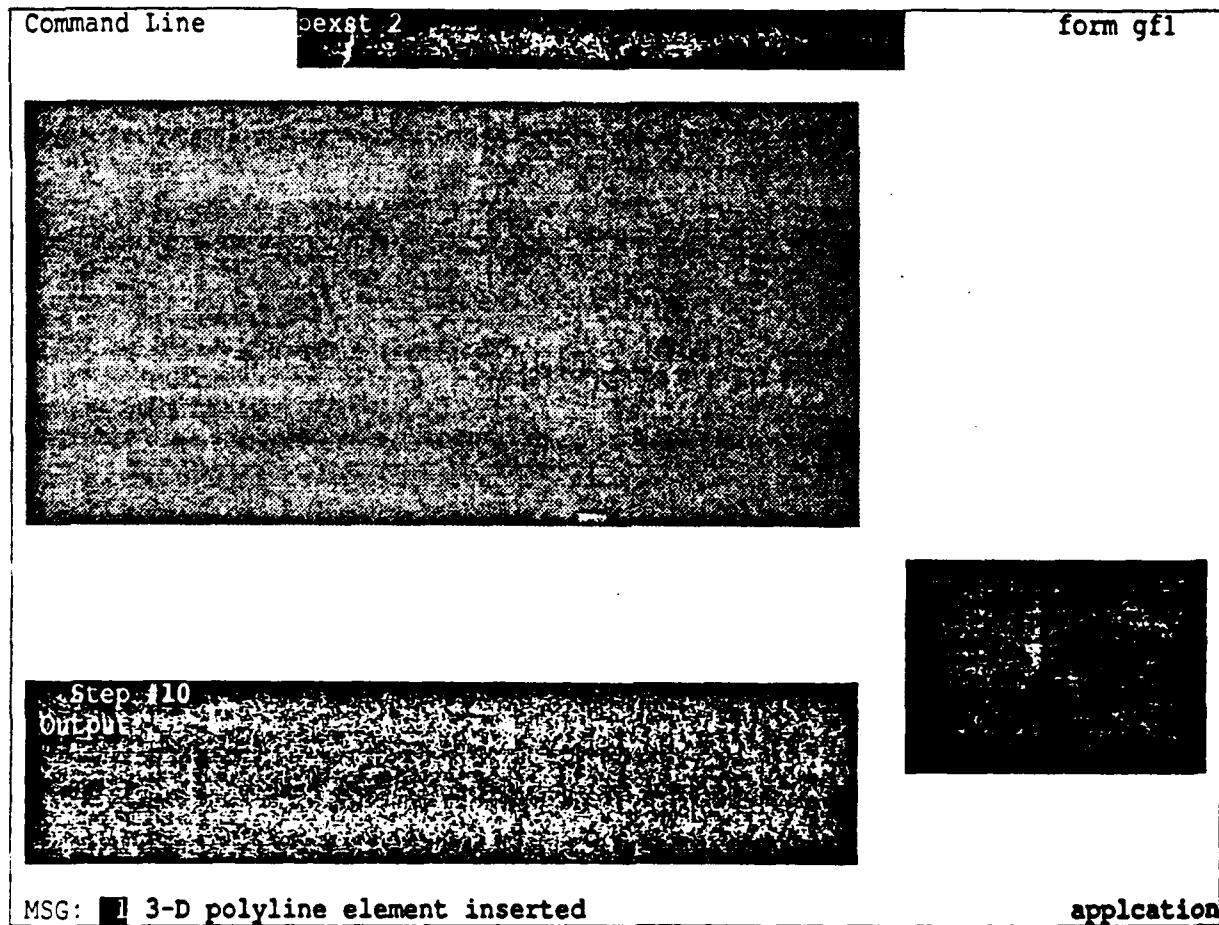
Figure 5-12a   (BEFORE)

Figure 5-12b   (AFTER)
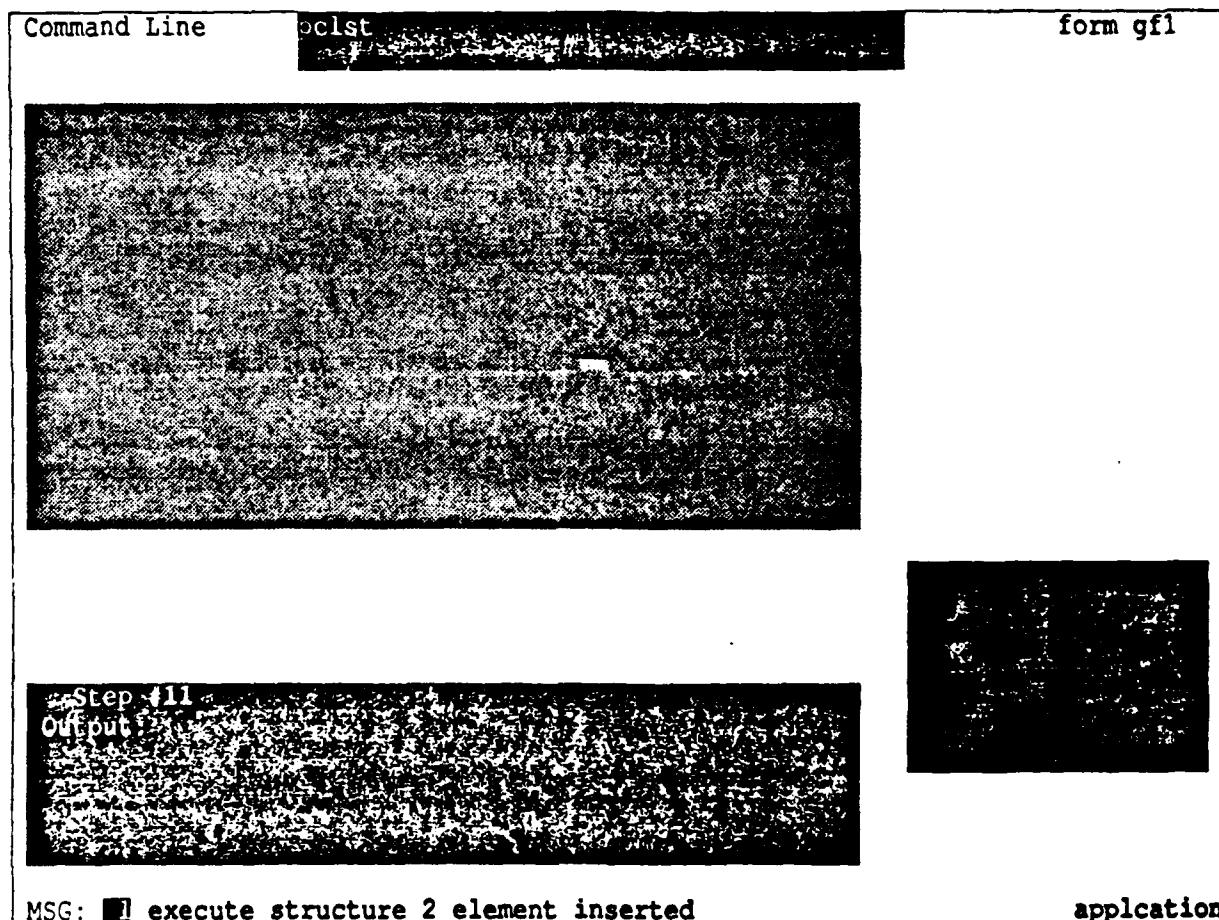
```
Command Line          clst                              form gf1




                                        Step #11
                                        Output:



MSG:  █ execute structure 2 element inserted            applcation
```

Figure 5-13a   (BEFORE)

Figure 5-13b   (AFTER)

Command Line | ɔopst 2 | form gfl

Step 12
Output

MSG: ▮ structure closed | applcation
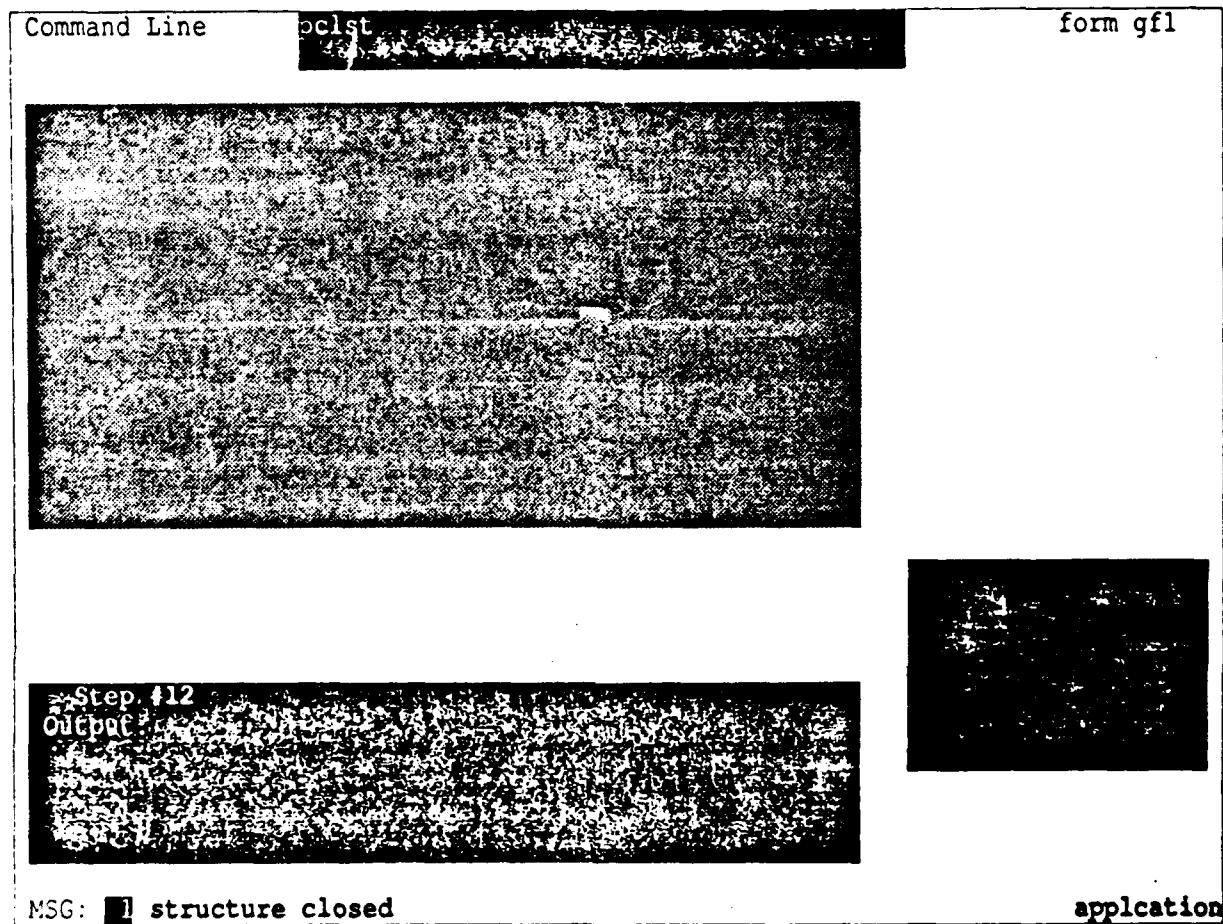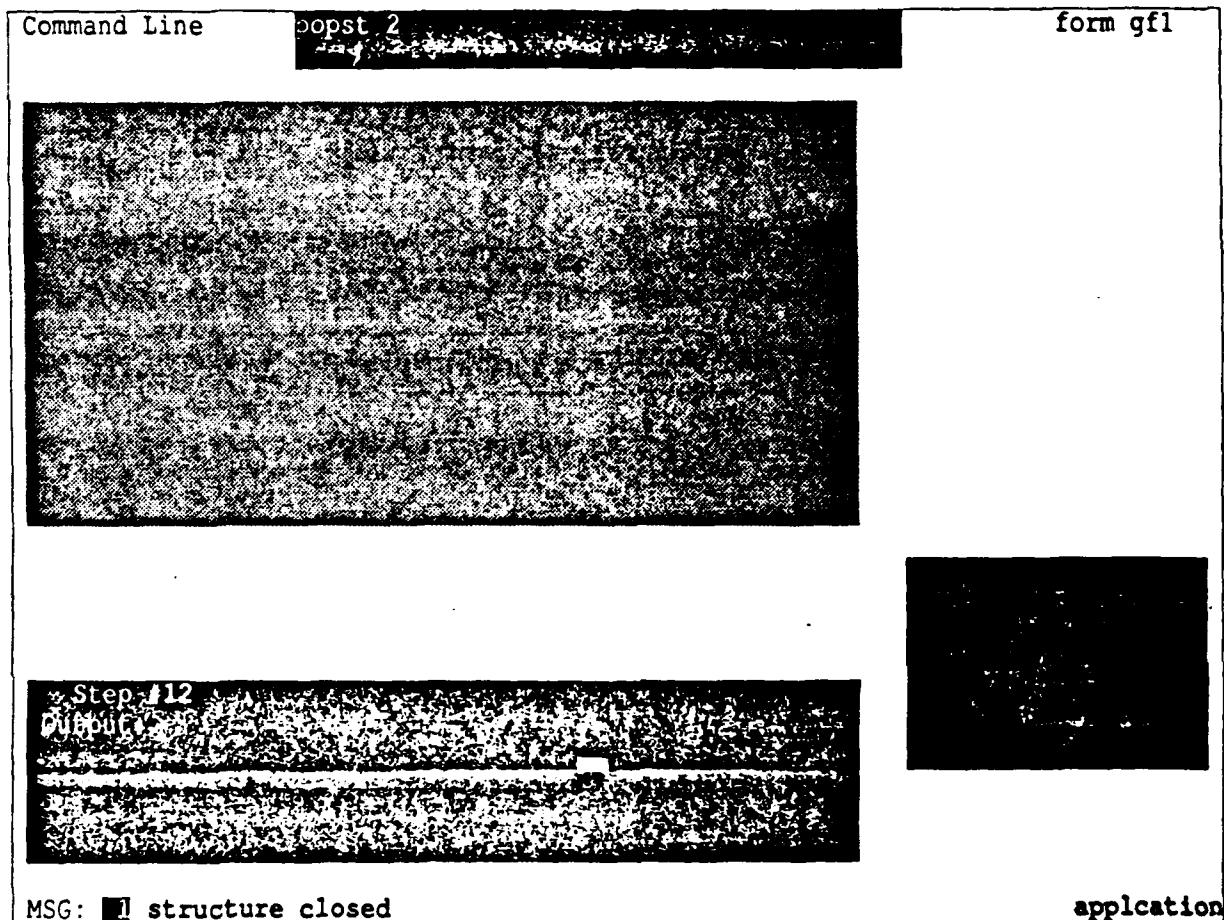
Figure 5-14a   (BEFORE)

Figure 5-14b   (AFTER)

Figure 5-15a   (BEFORE)

Command Line   ostxci blue   form gfl

Step #14
Output:

MSG: text color set to BLUE   applcation

Figure 5-15b   (AFTER)

Figure 5-16a   (BEFORE)

Command Line   Jschh .1   form gfl

Step #15
Output:

MSG: ▉] character height set to 0.1 of nominal   applcation
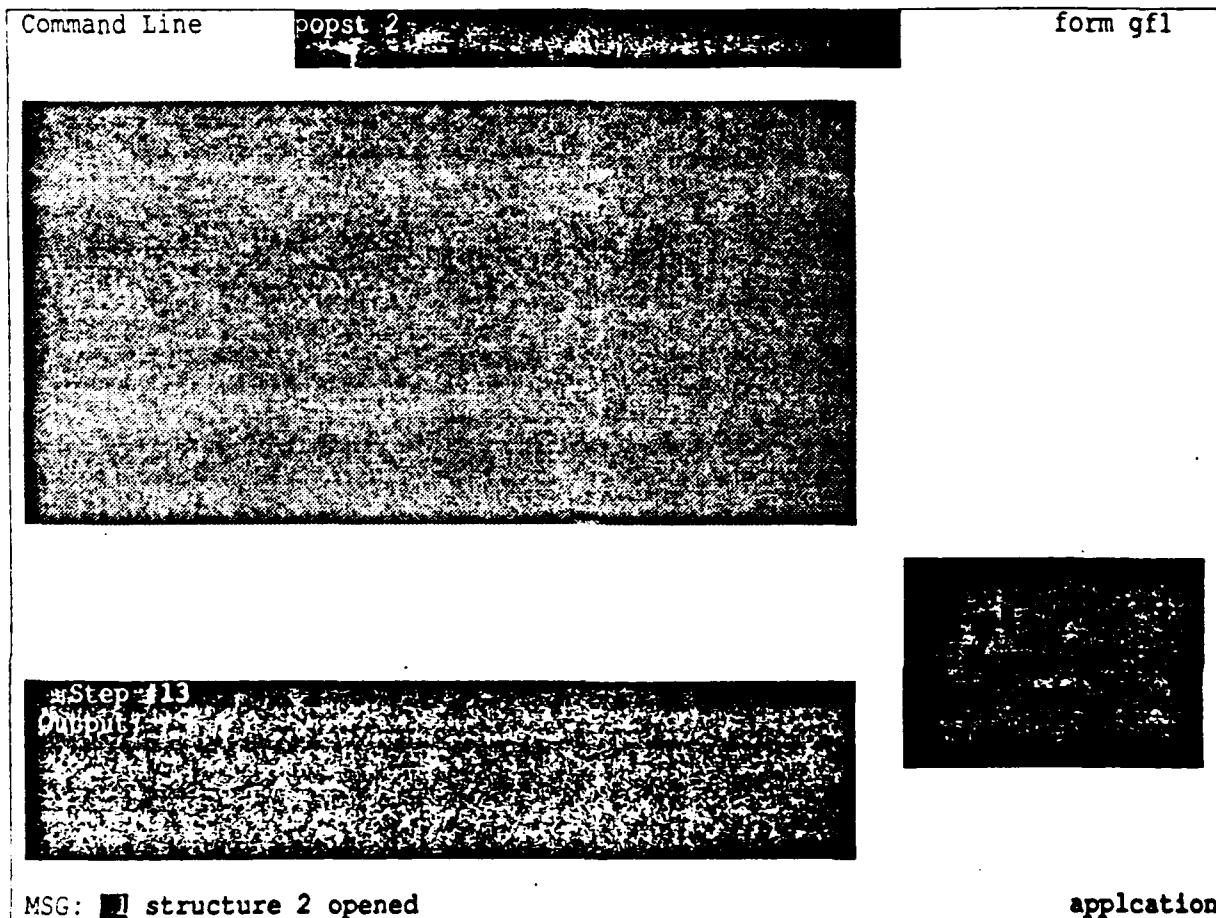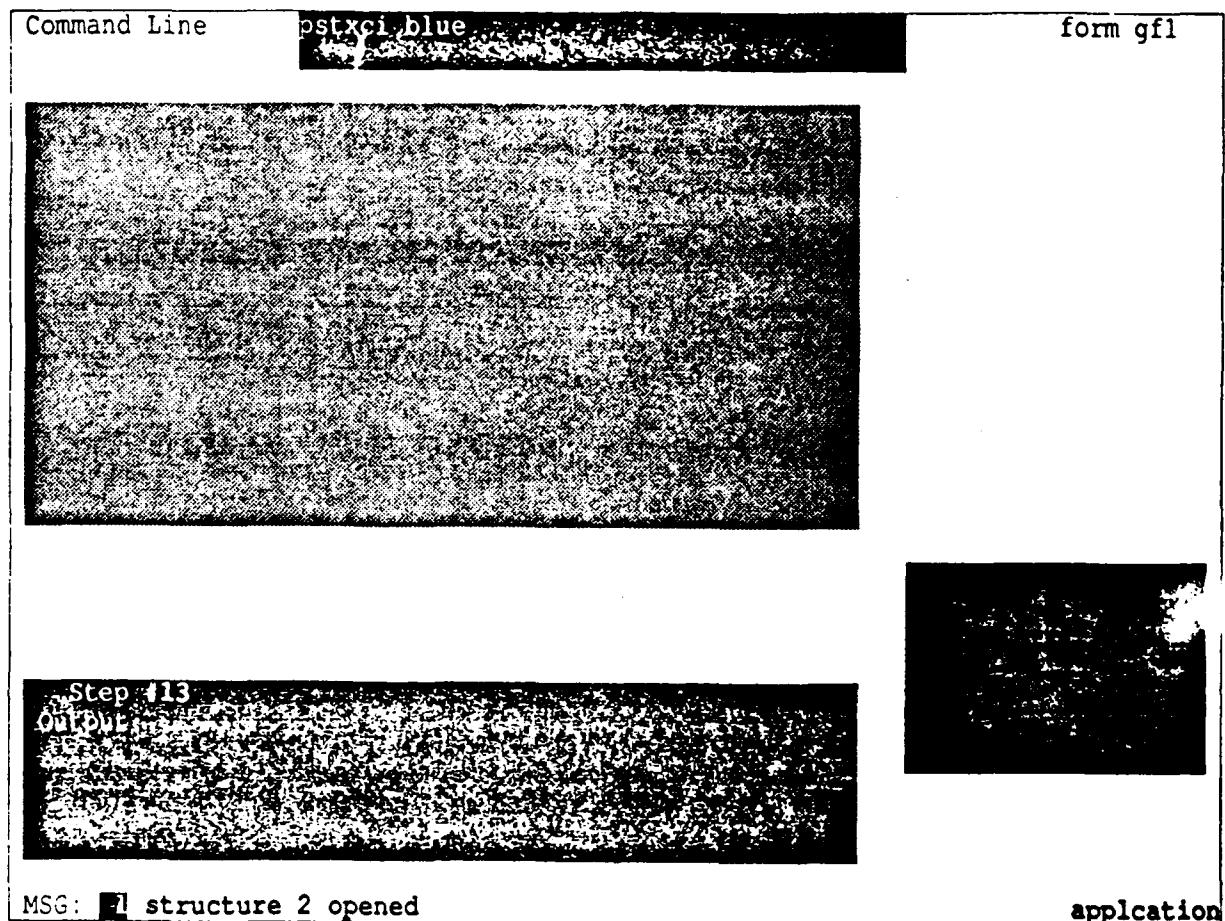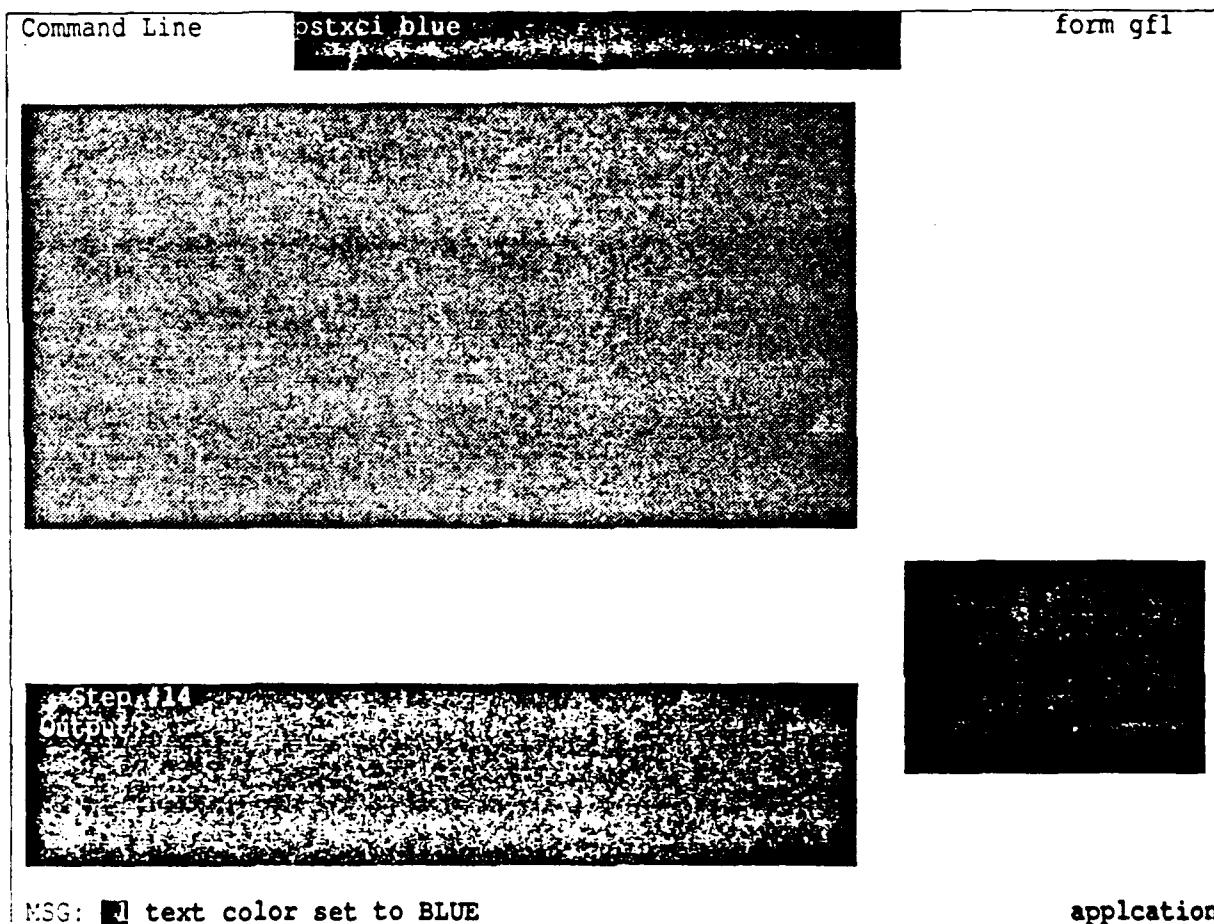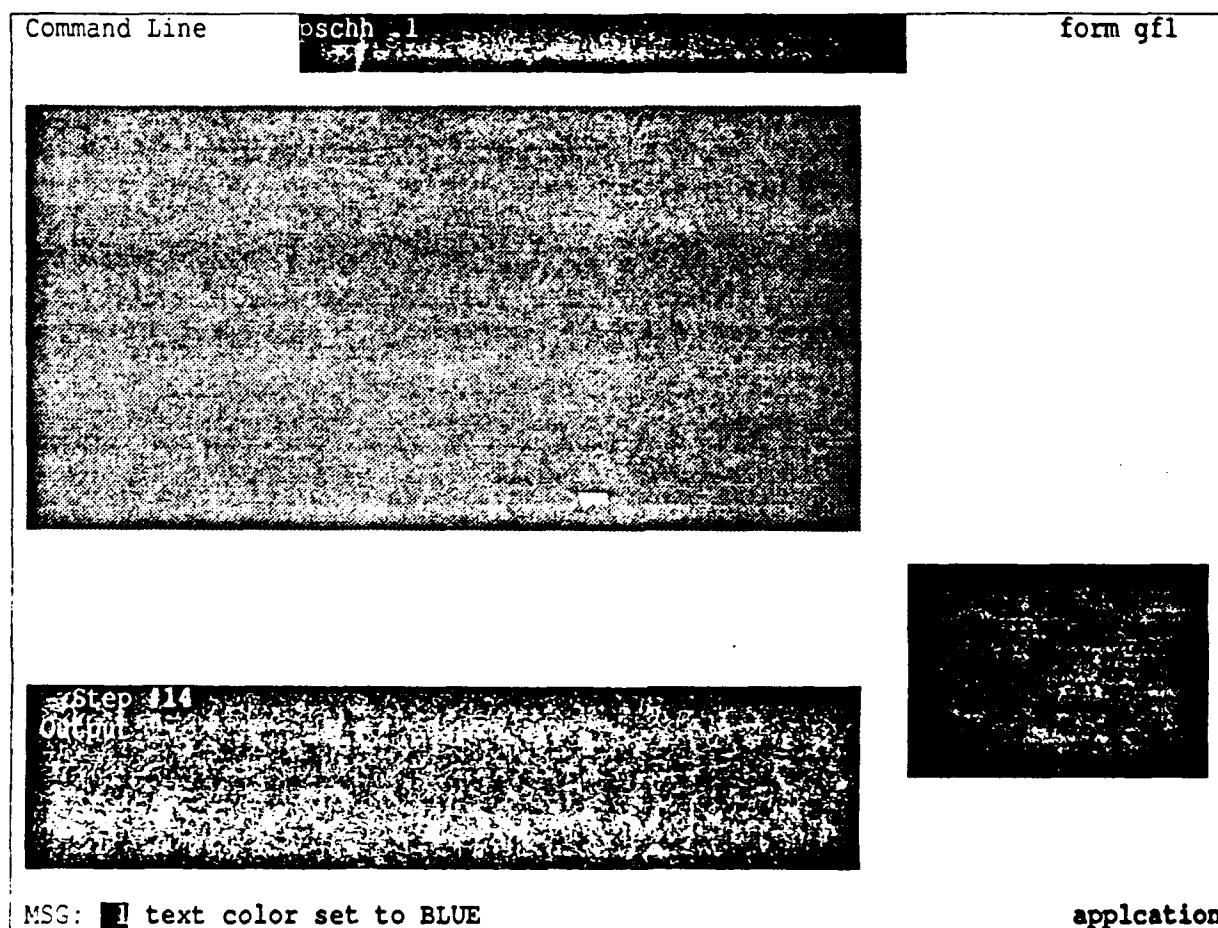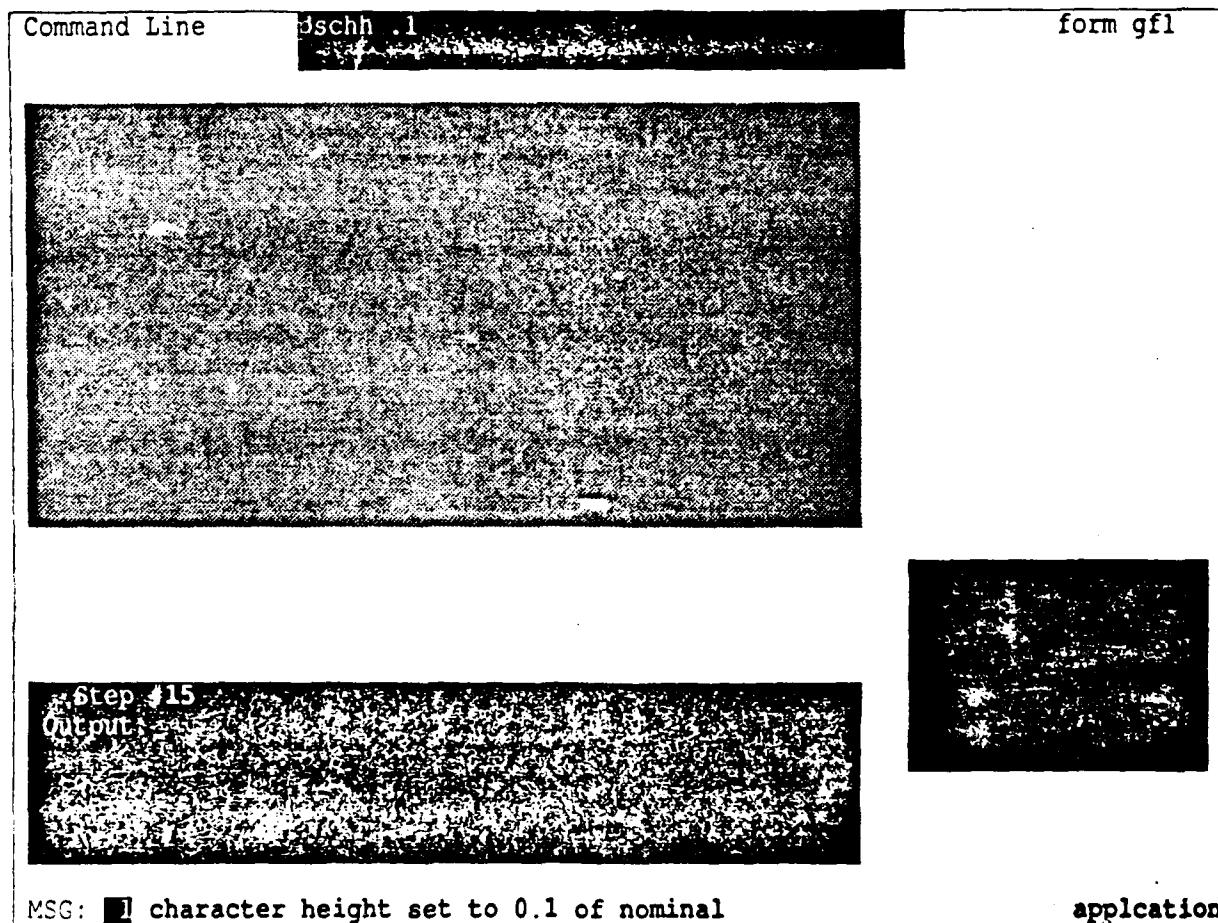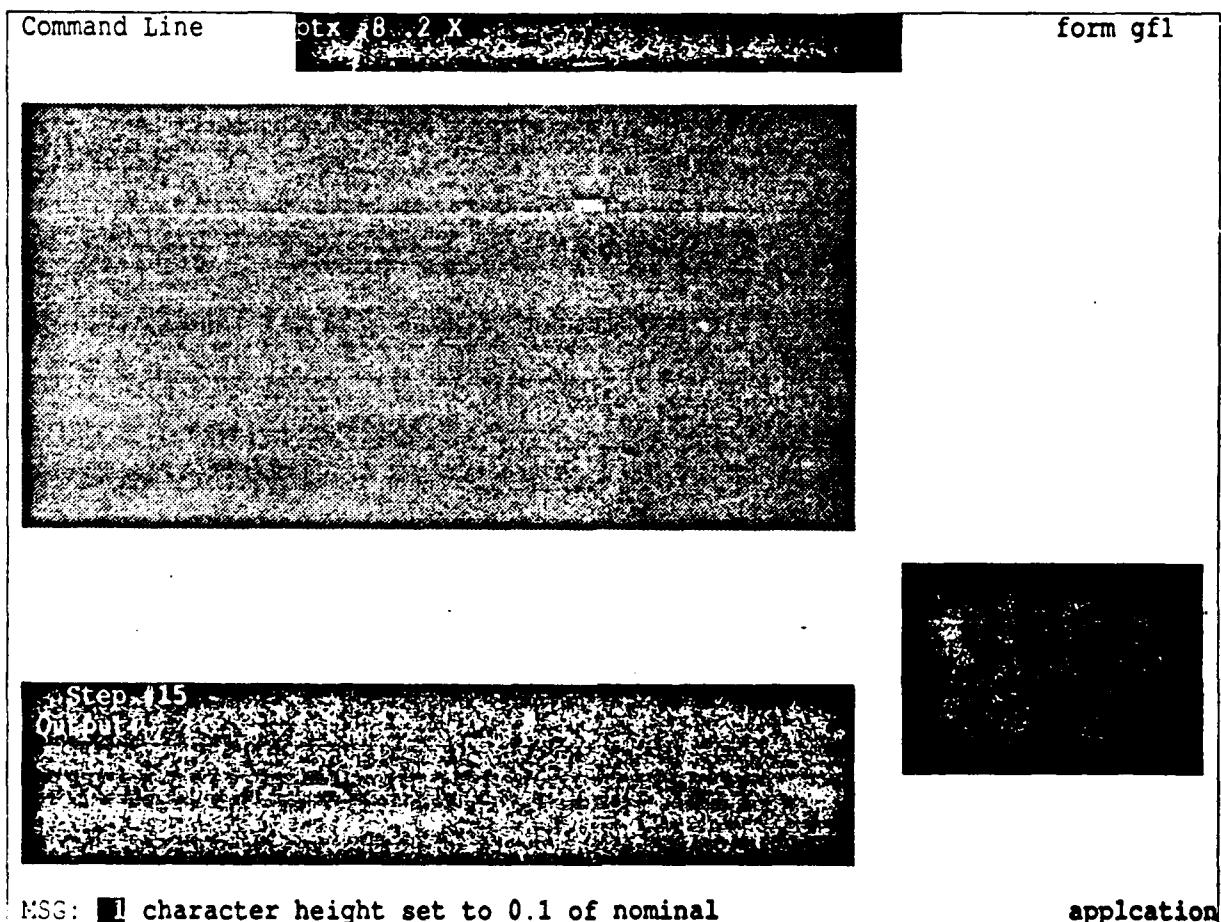
Figure 5-16b   (AFTER)

Figure 5-17a   (BEFORE)
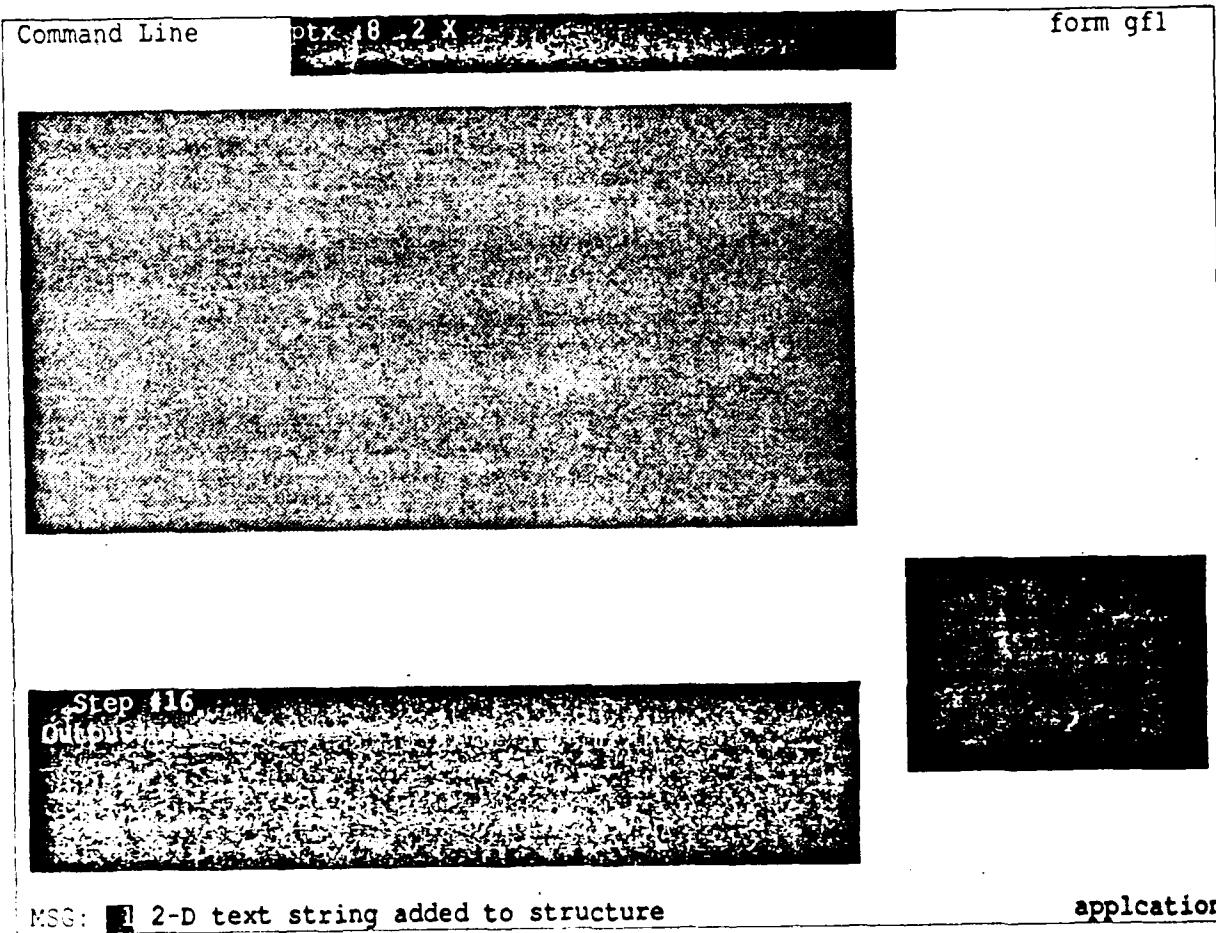
Figure 5-17b   (AFTER)
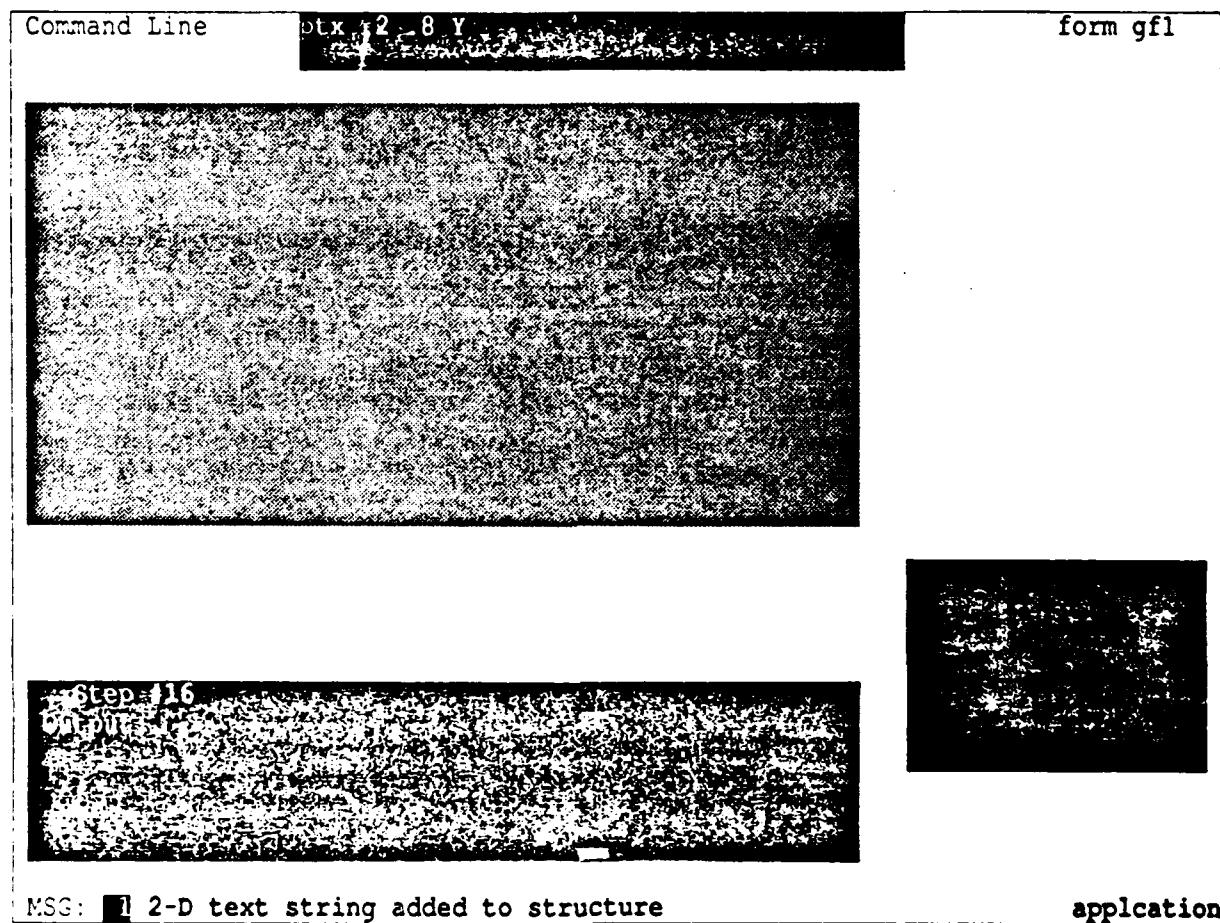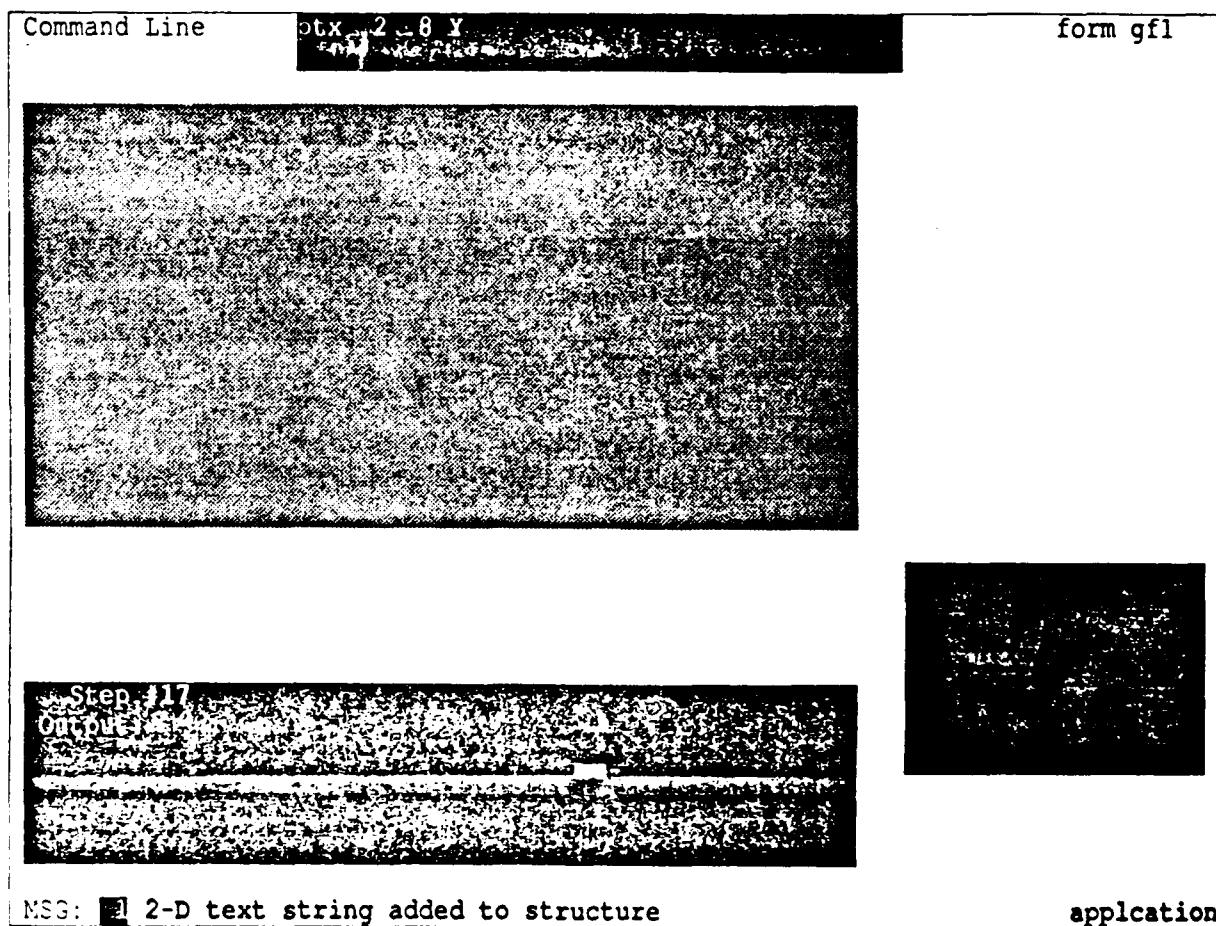
Figure 5-18a   (BEFORE)

Figure 5-18b   (AFTER)
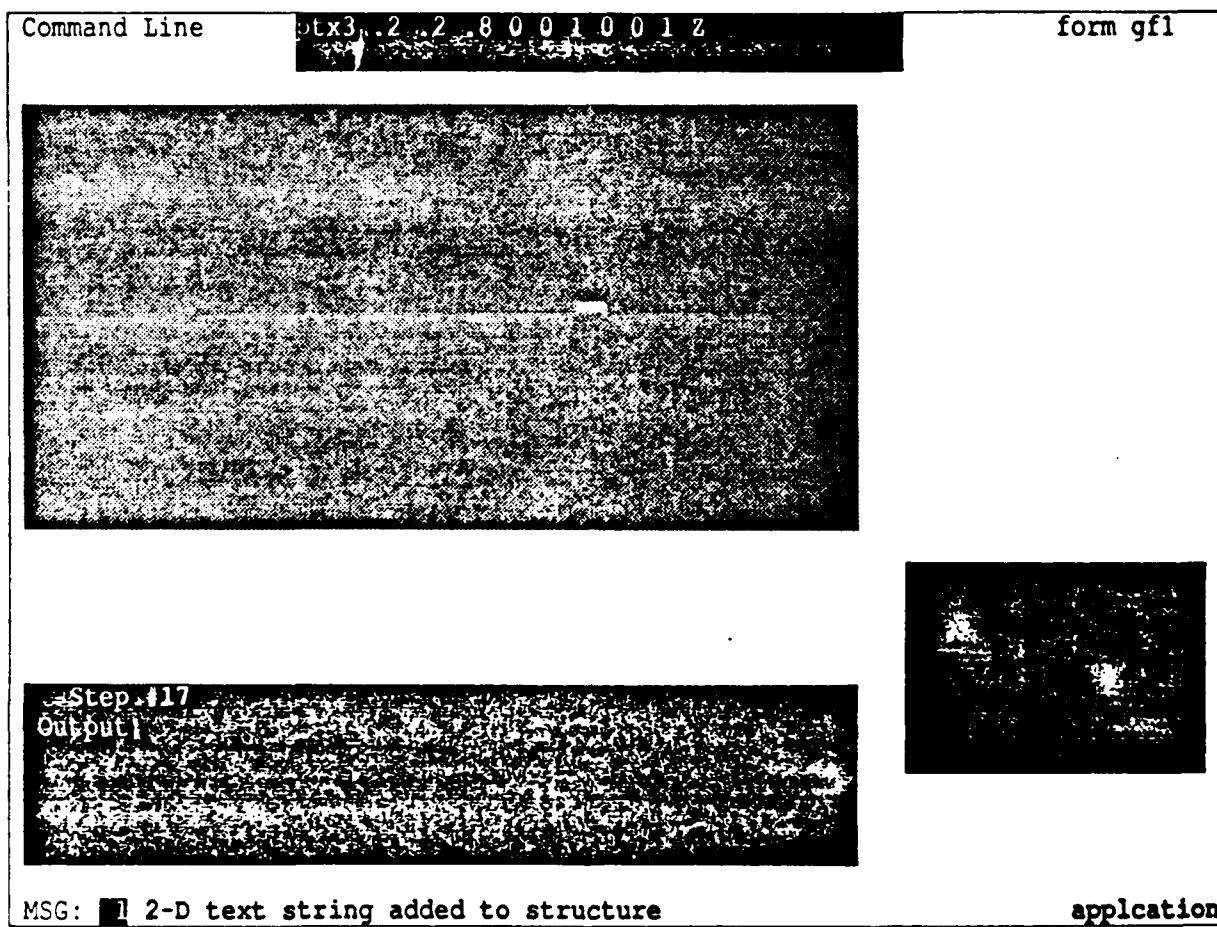
Figure 5-19a   (BEFORE)
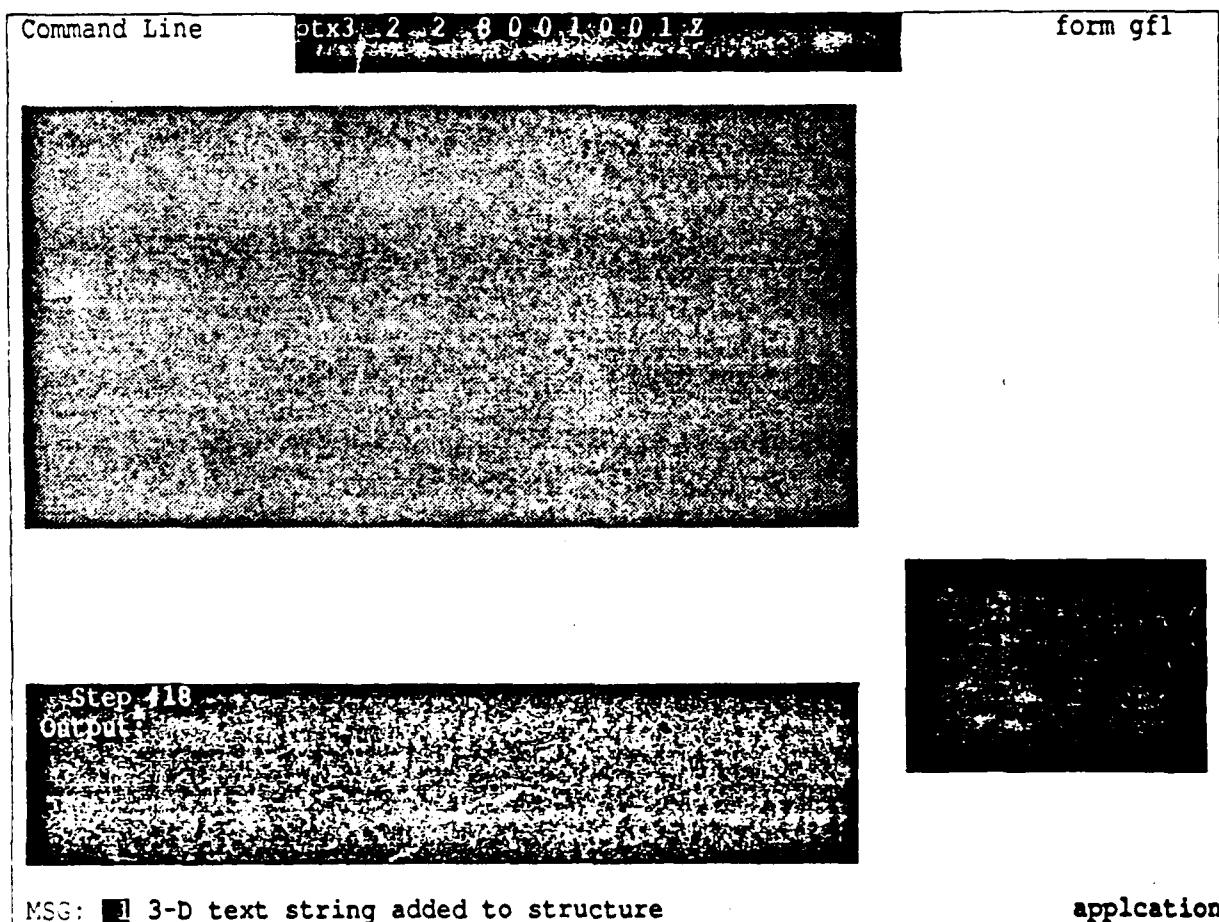
Figure 5-19b   (AFTER)

Command Line   pclst   form gfl

Step #18
output:

MSG: ▊ 3-D text string added to structure   applcation

Figure 5-20a   (BEFORE)

Command Line   bclst   form

MSG: ▓ structure closed   applcation
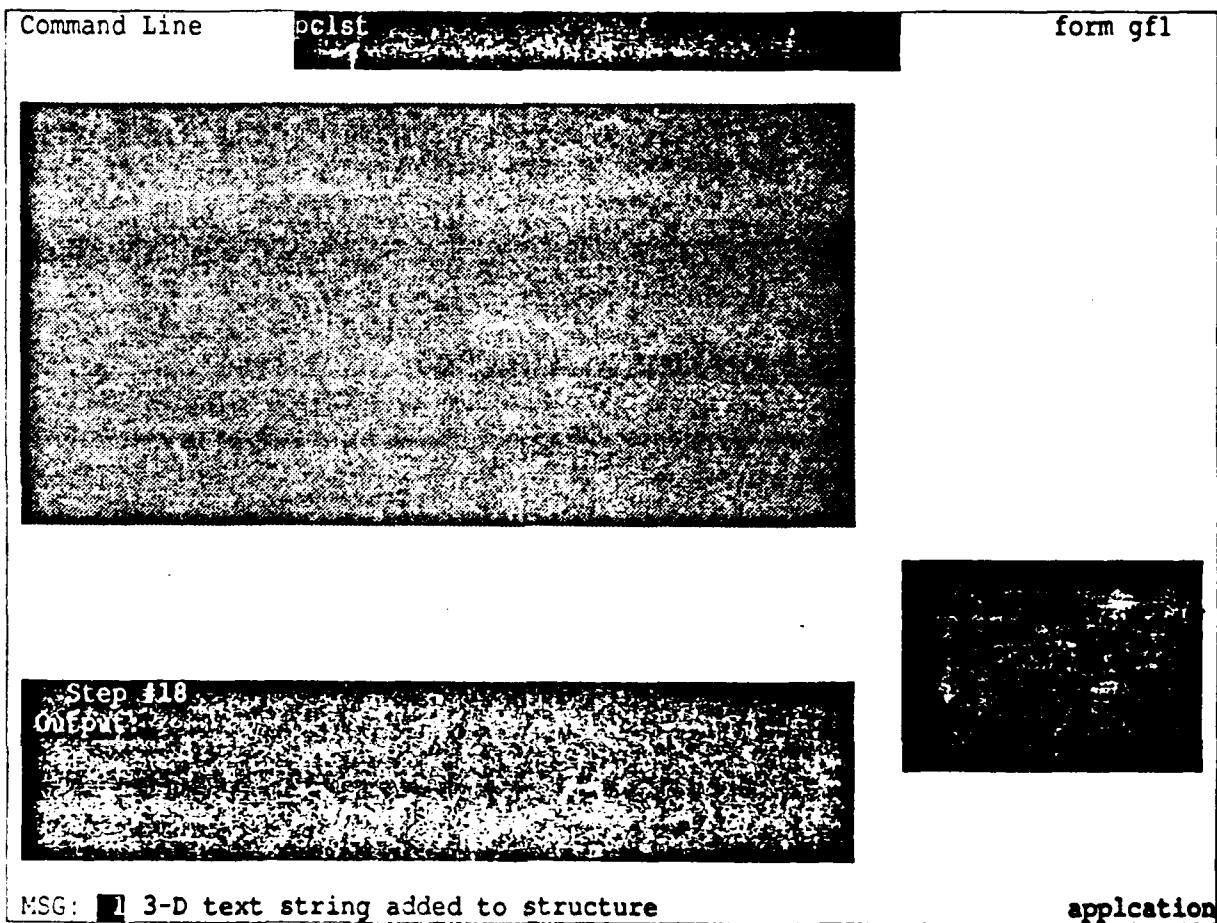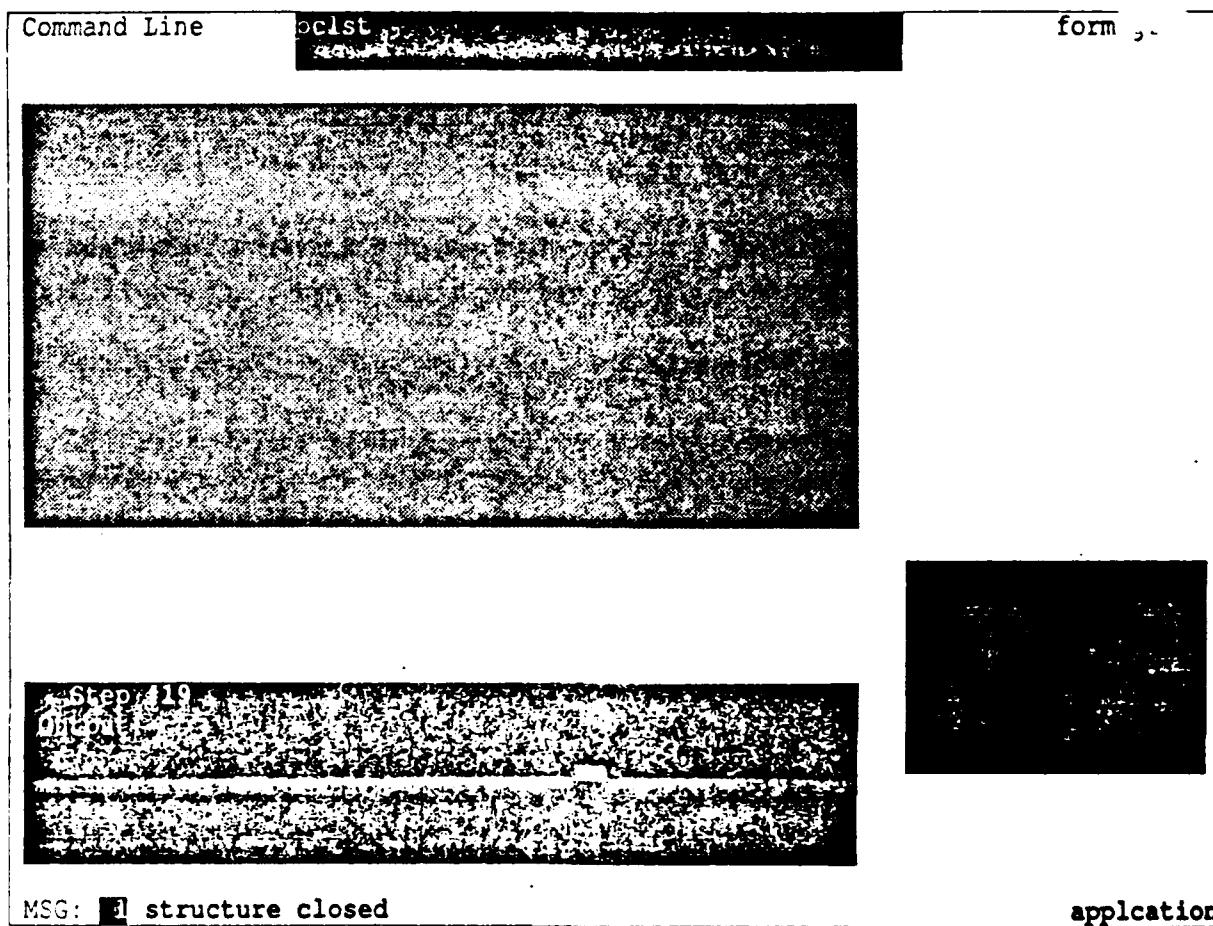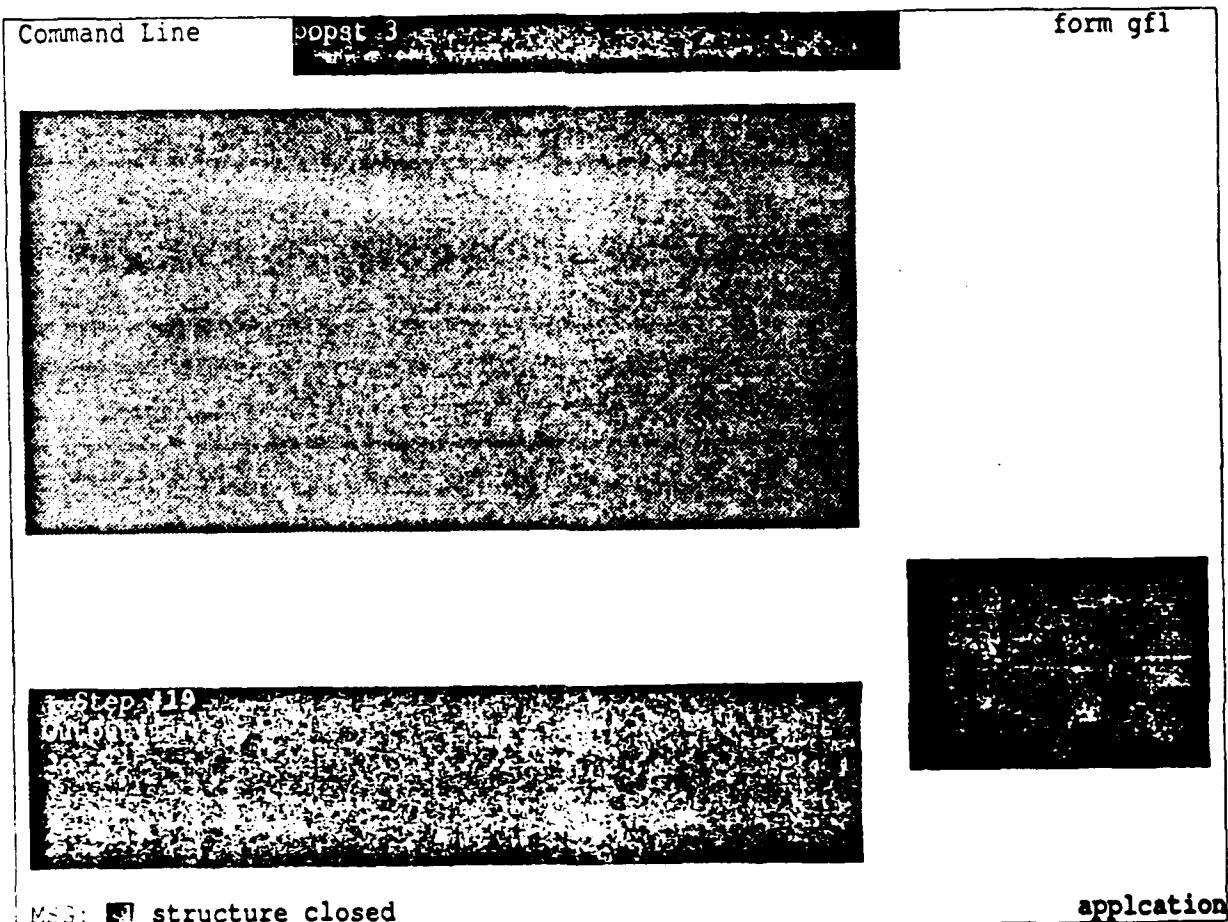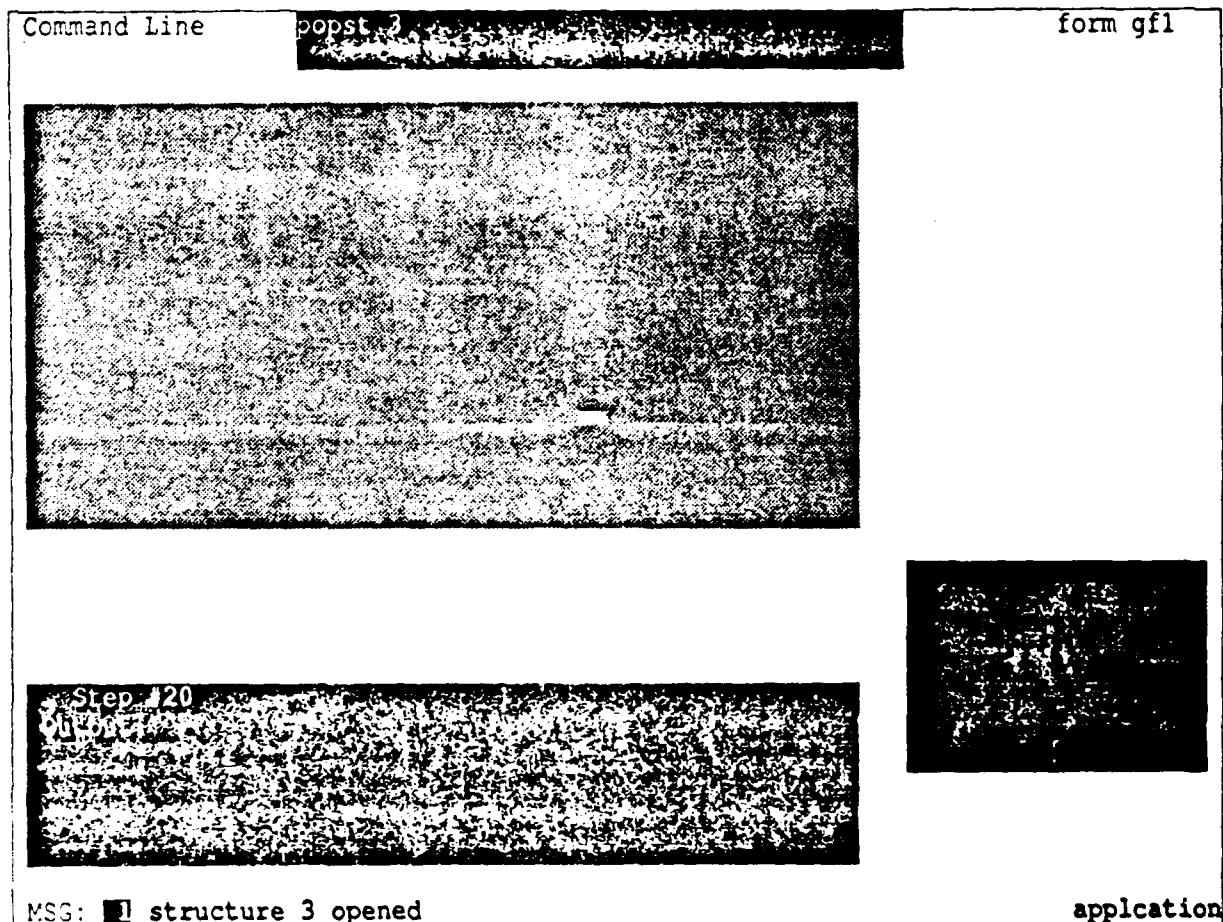
Figure 5-20b   (AFTER)

Figure 5-21a   (BEFORE)

Figure 5-21b   (AFTER)
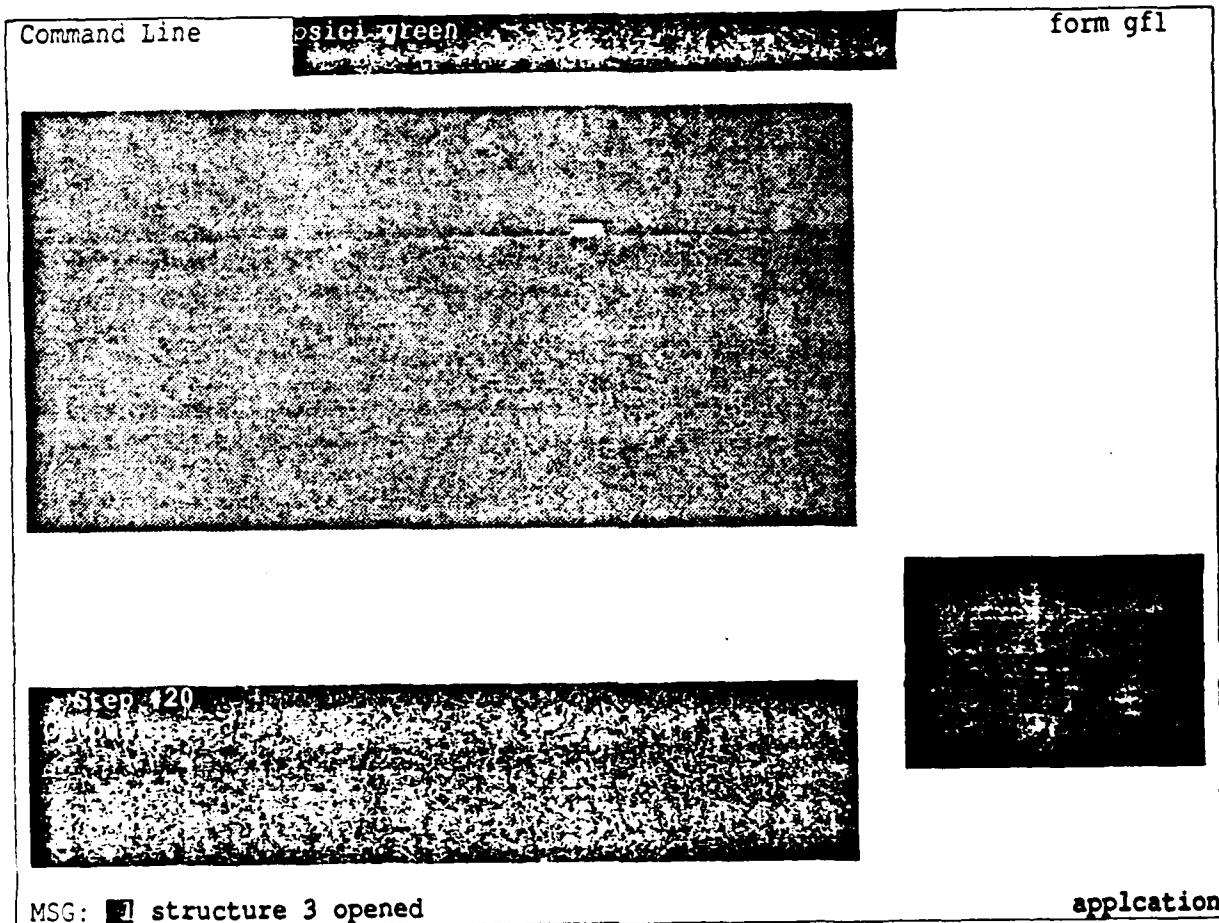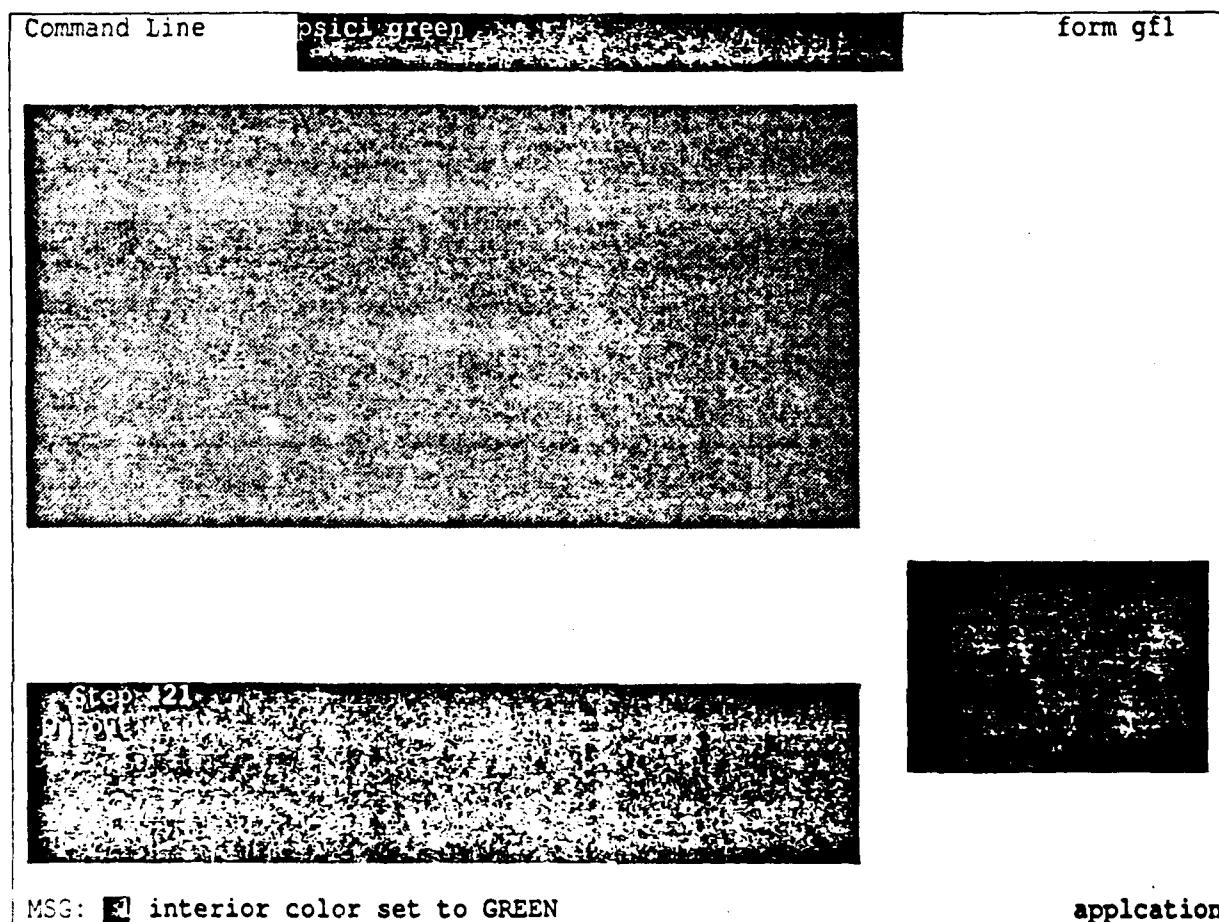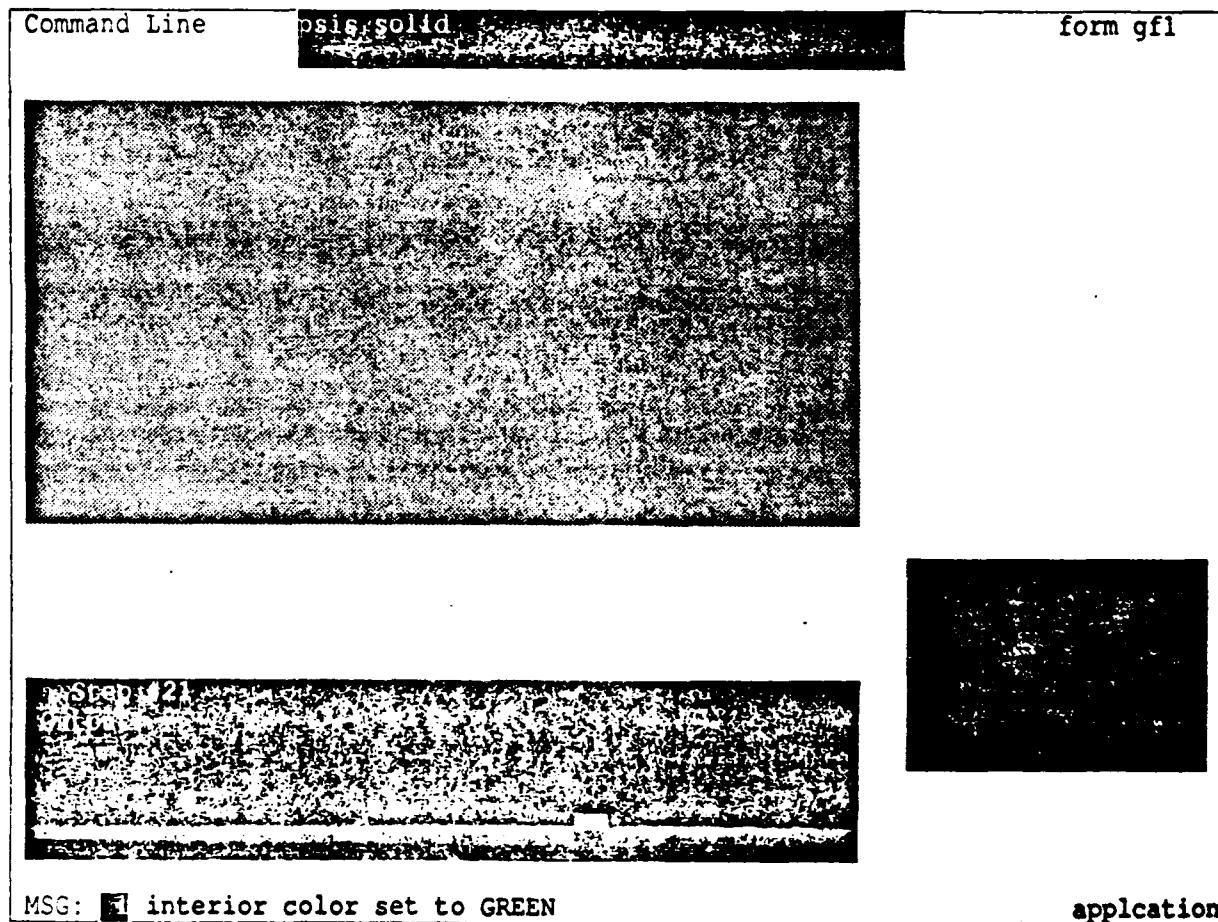
Figure 5-22a   (BEFORE)

Figure 5-22b (AFTER)

Figure 5-23a   (BEFORE)
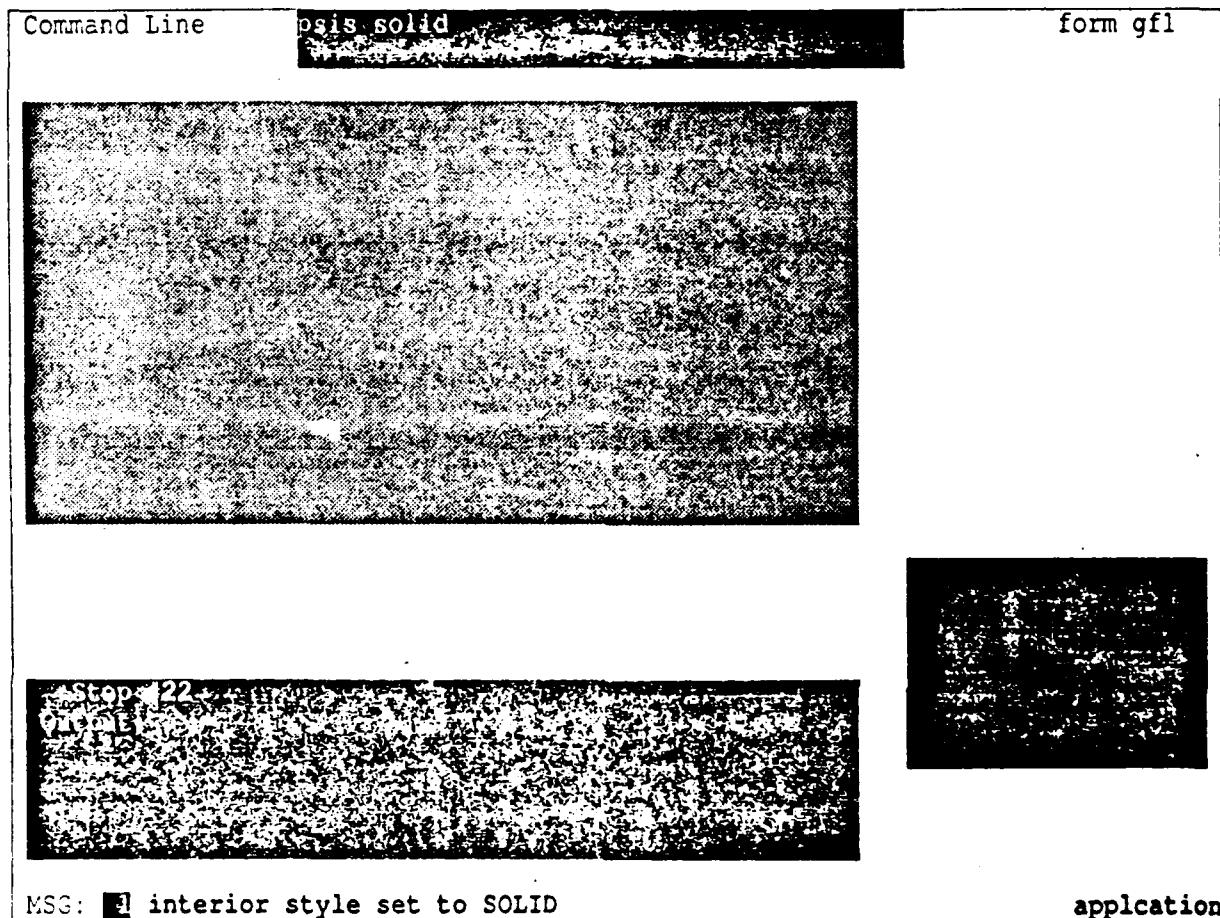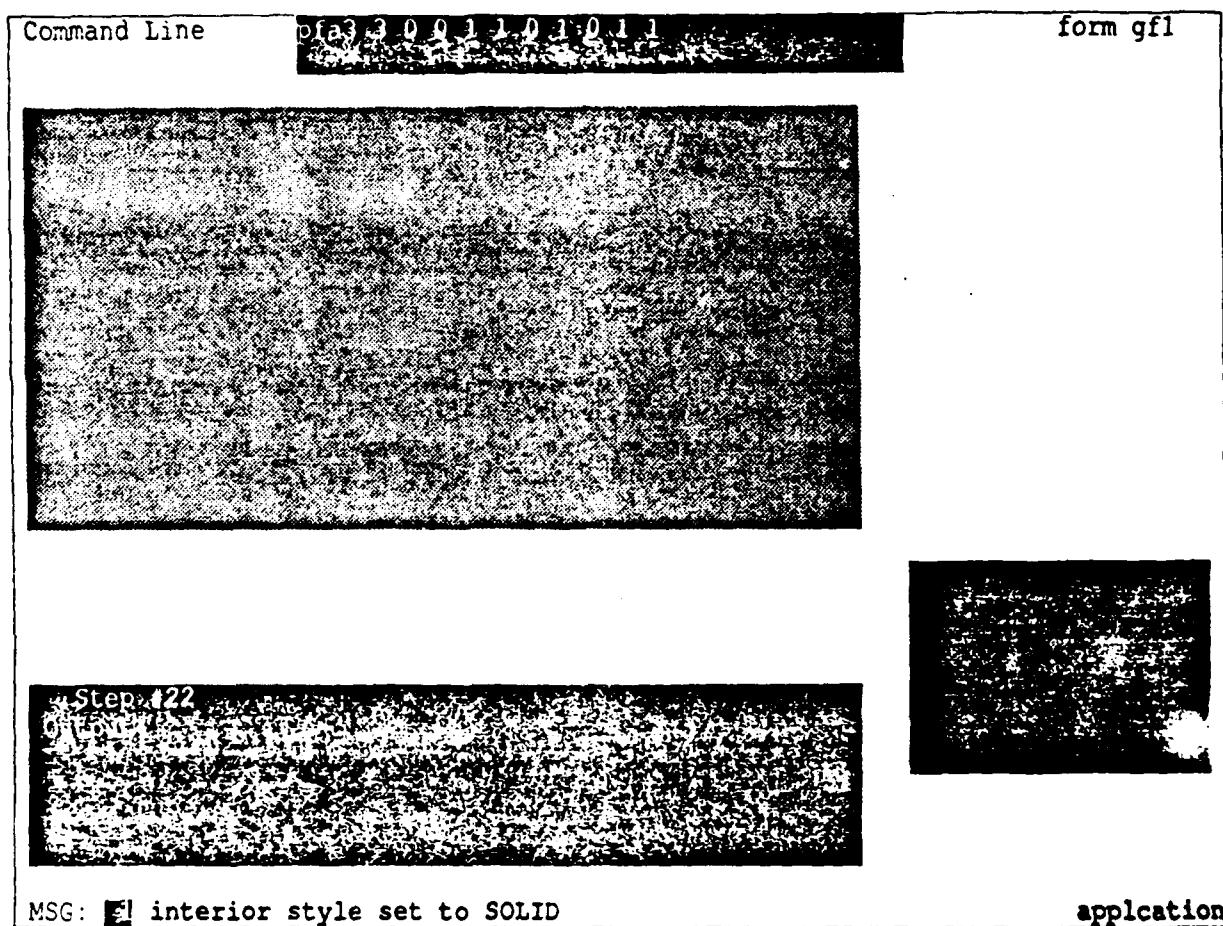
Figure 5-23b   (AFTER)

Figure 5-24a   (BEFORE)
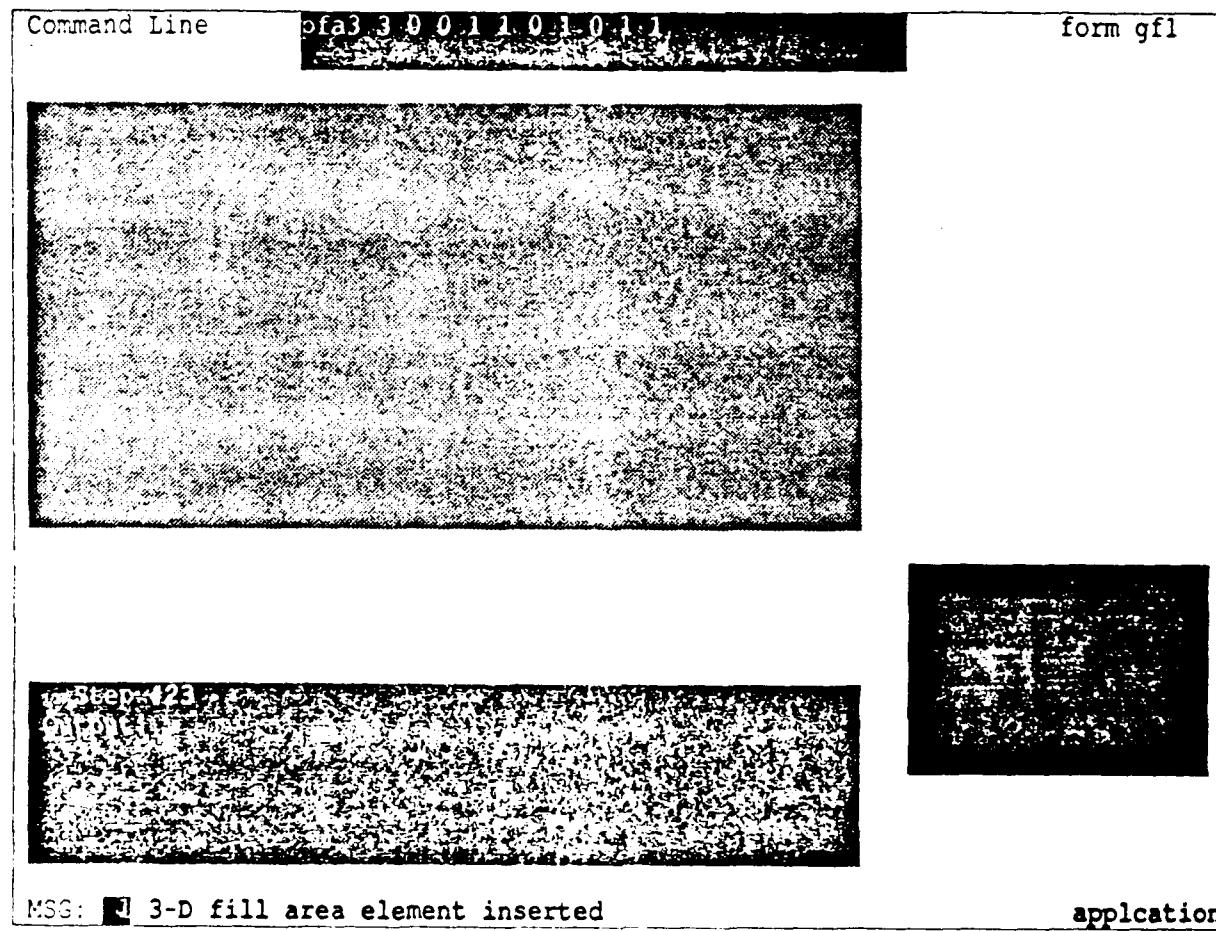
Figure 5-24b   (AFTER)
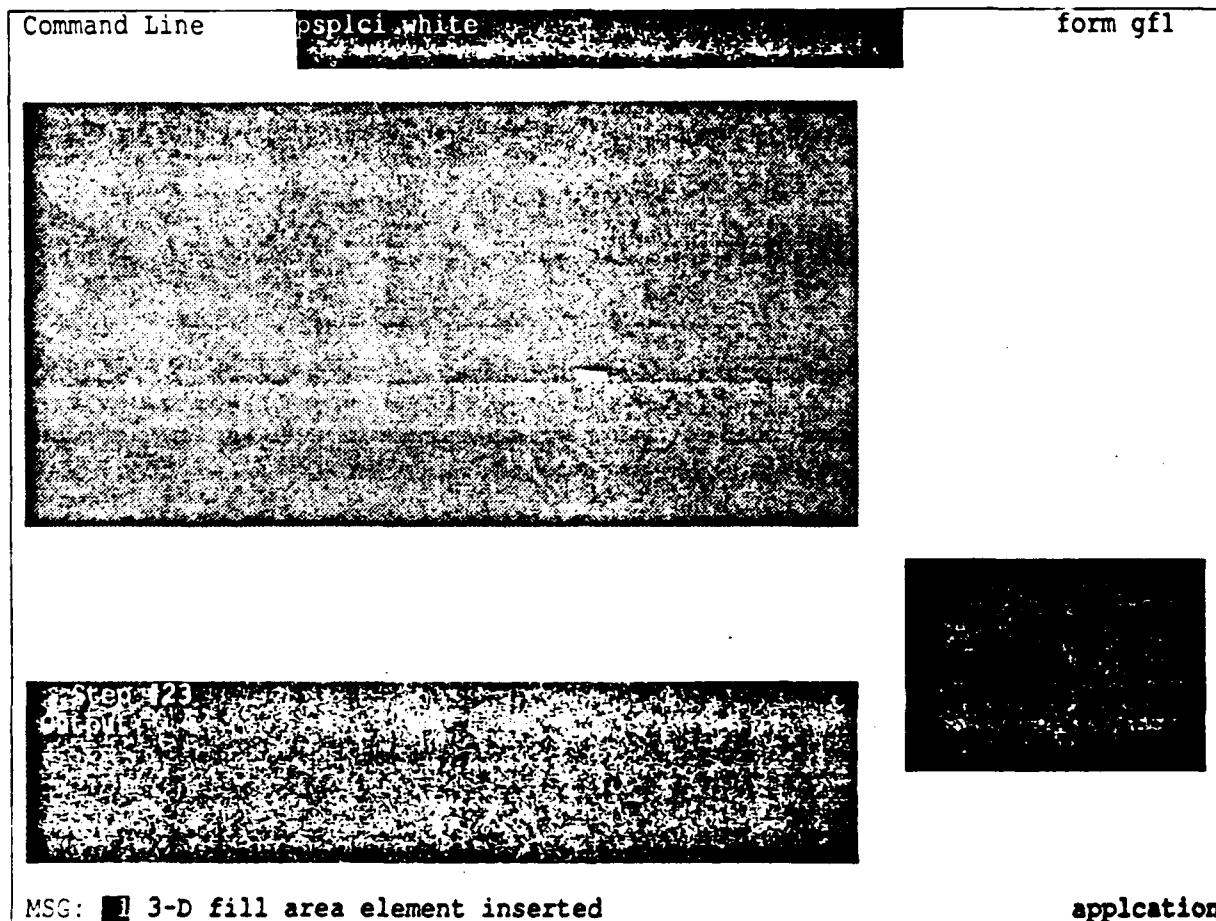
Figure 5-25a   (BEFORE)
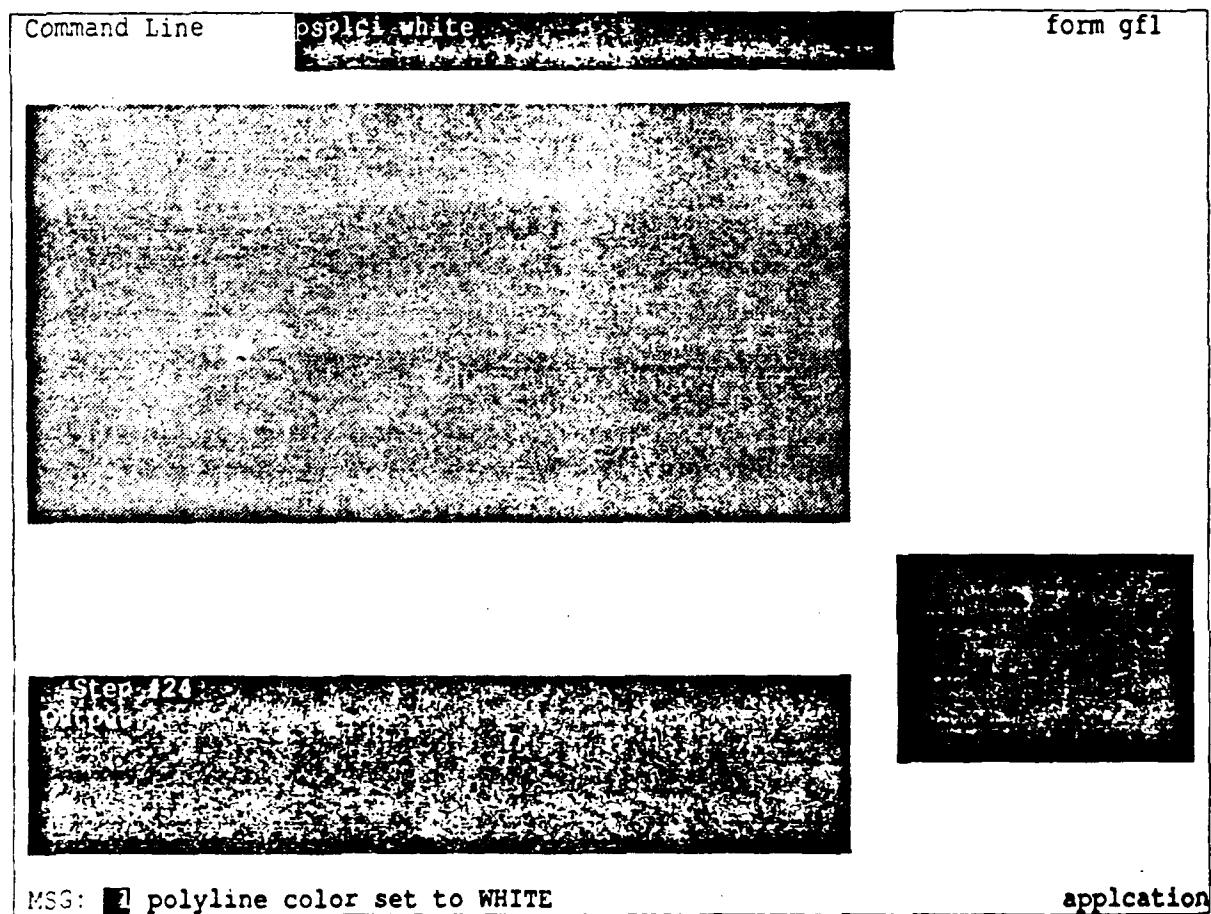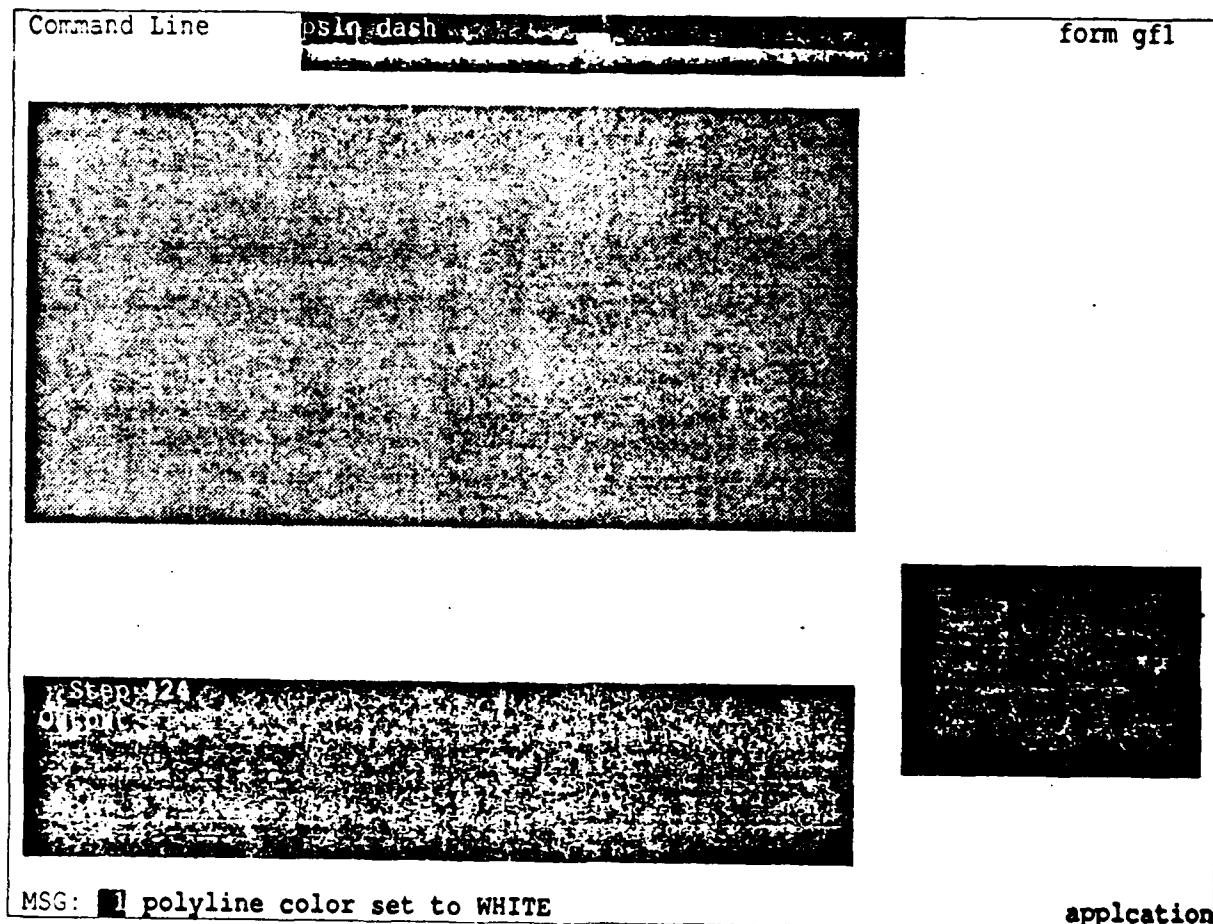
Figure 5-25b   (AFTER)
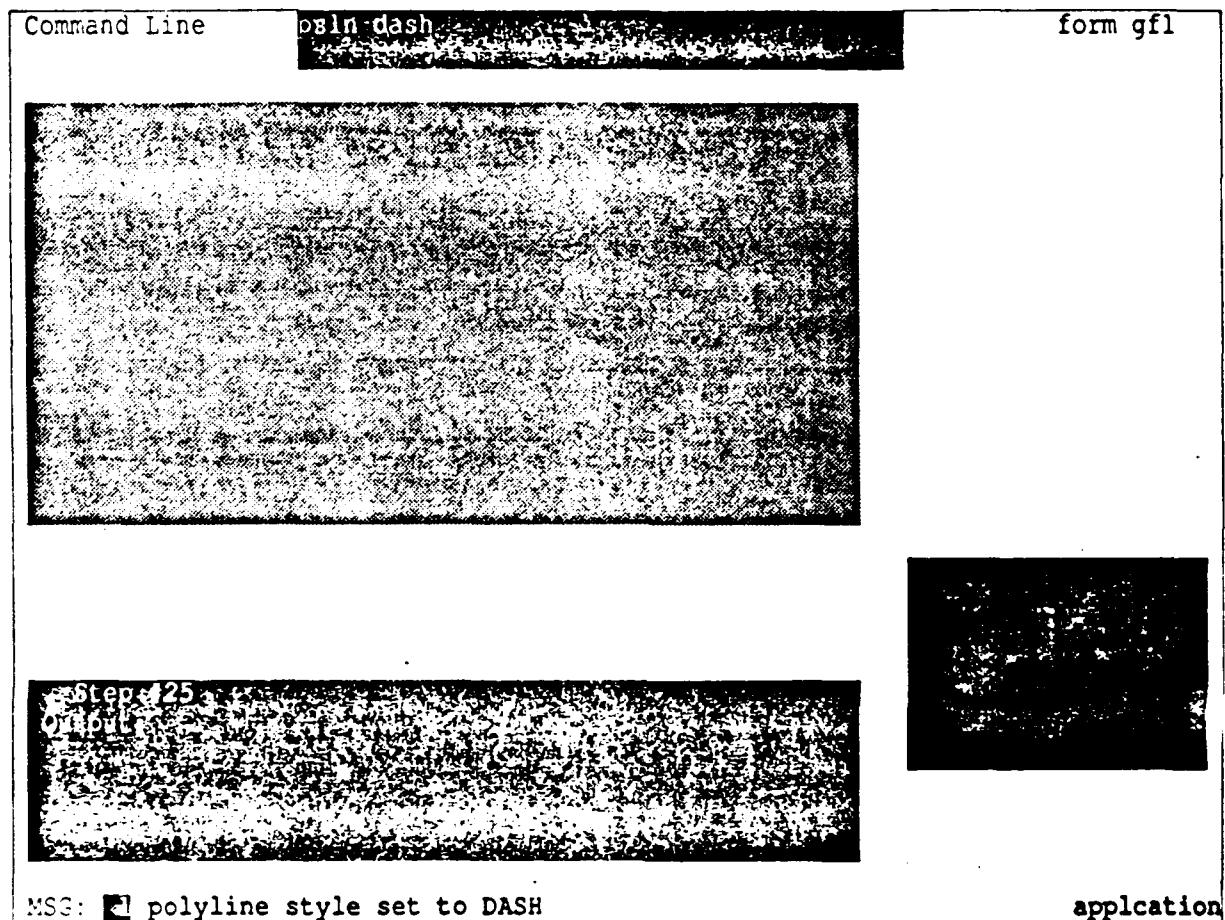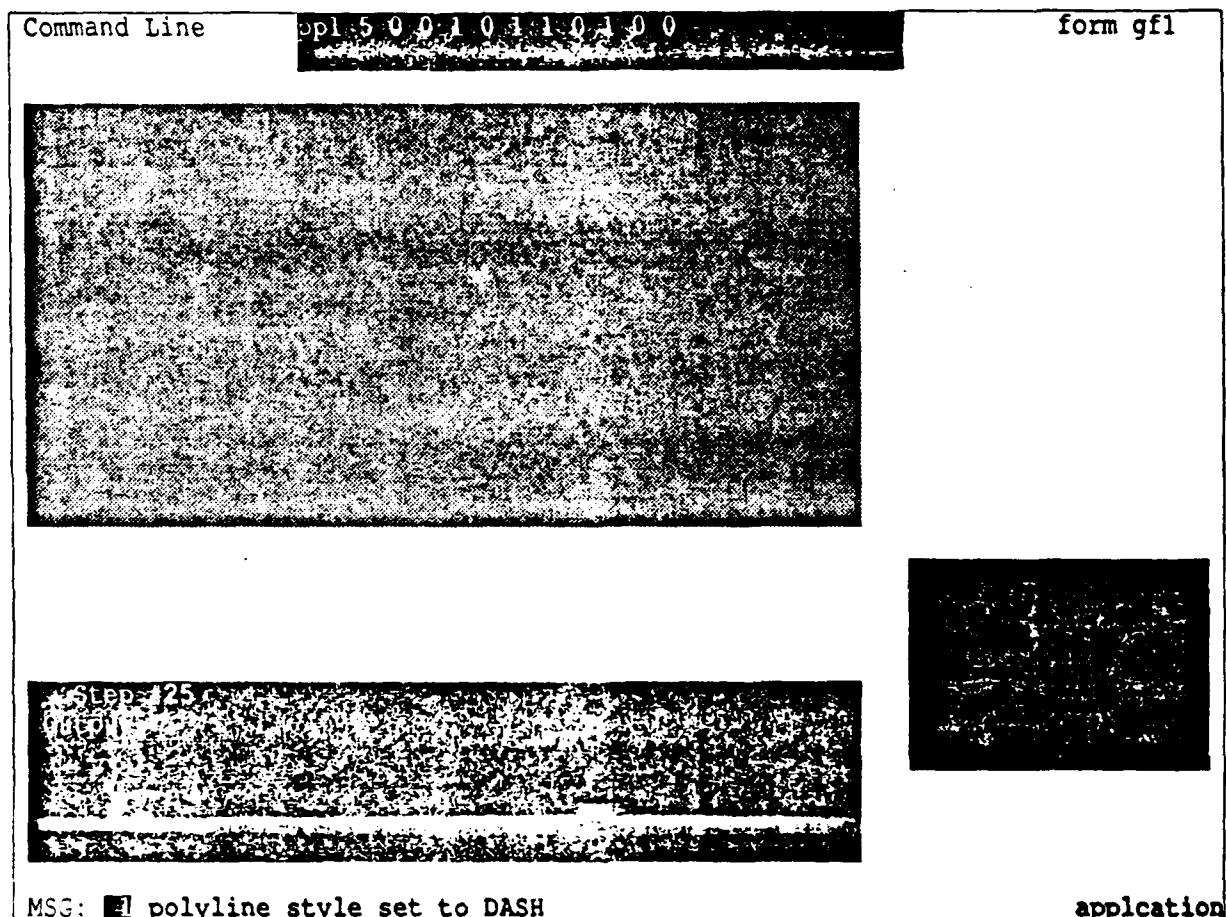
Figure 5-26a   (BEFORE)

Figure 5-26b   (AFTER)

Figure 5-27a   (BEFORE)
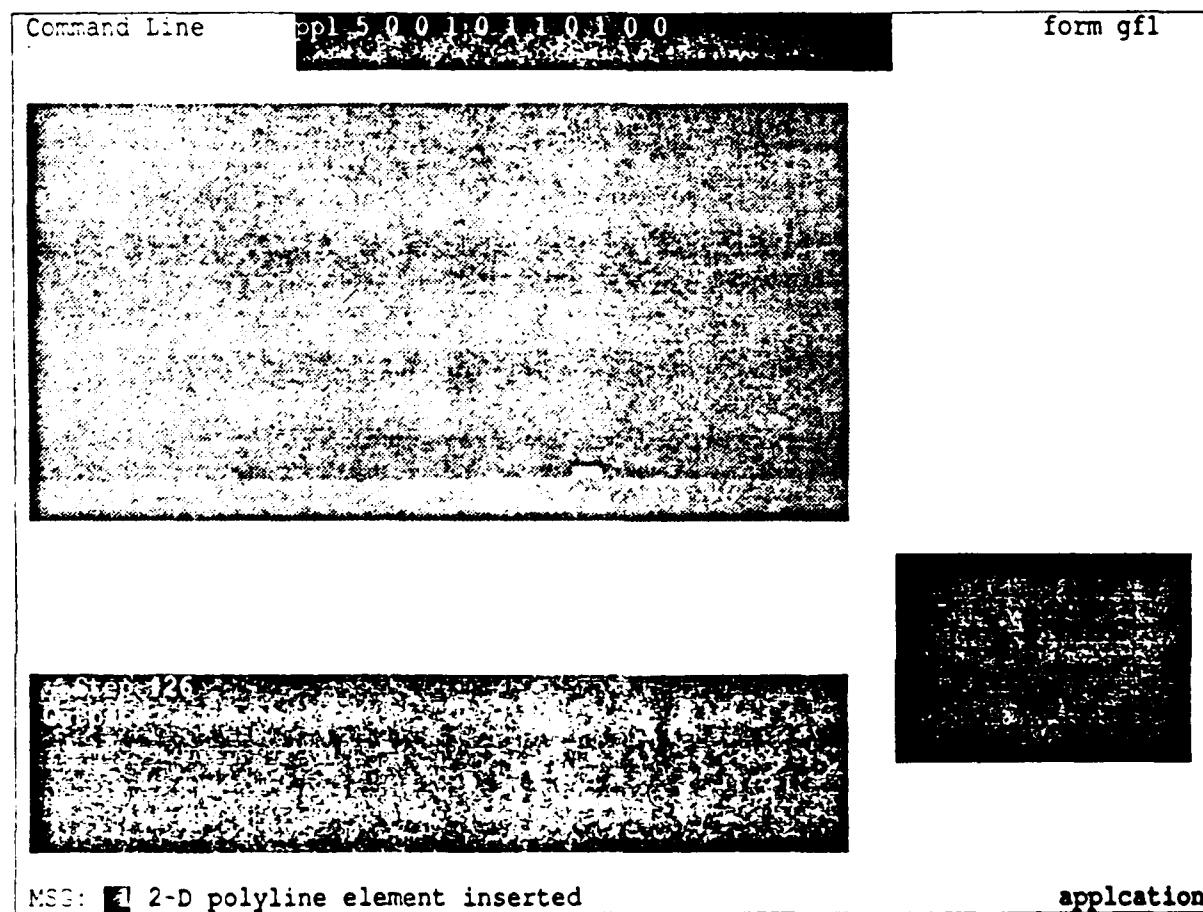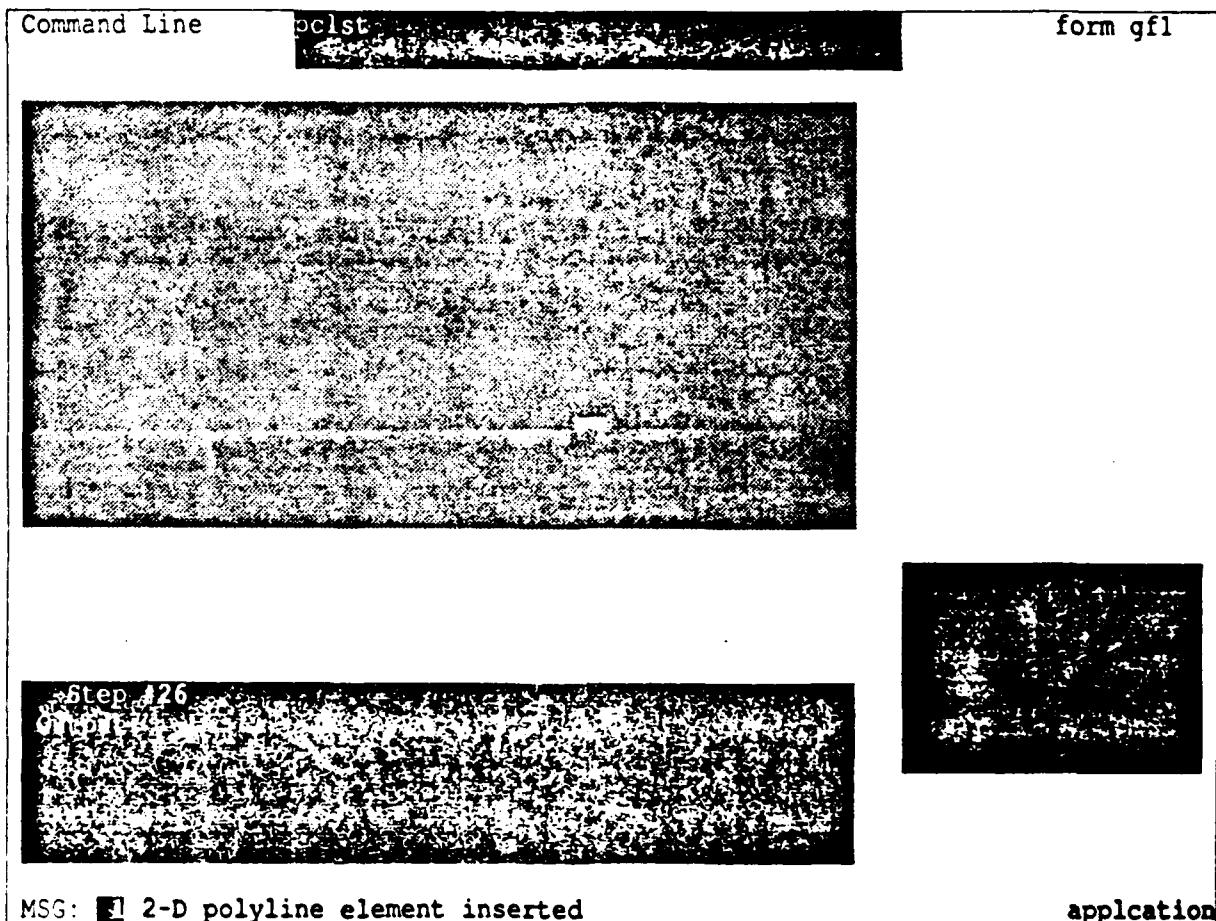
Figure 5-27b   (AFTER)

Figure 5-28a   (BEFORE)

Command Line   pclst   form gfl

MSG: structure closed   applcation
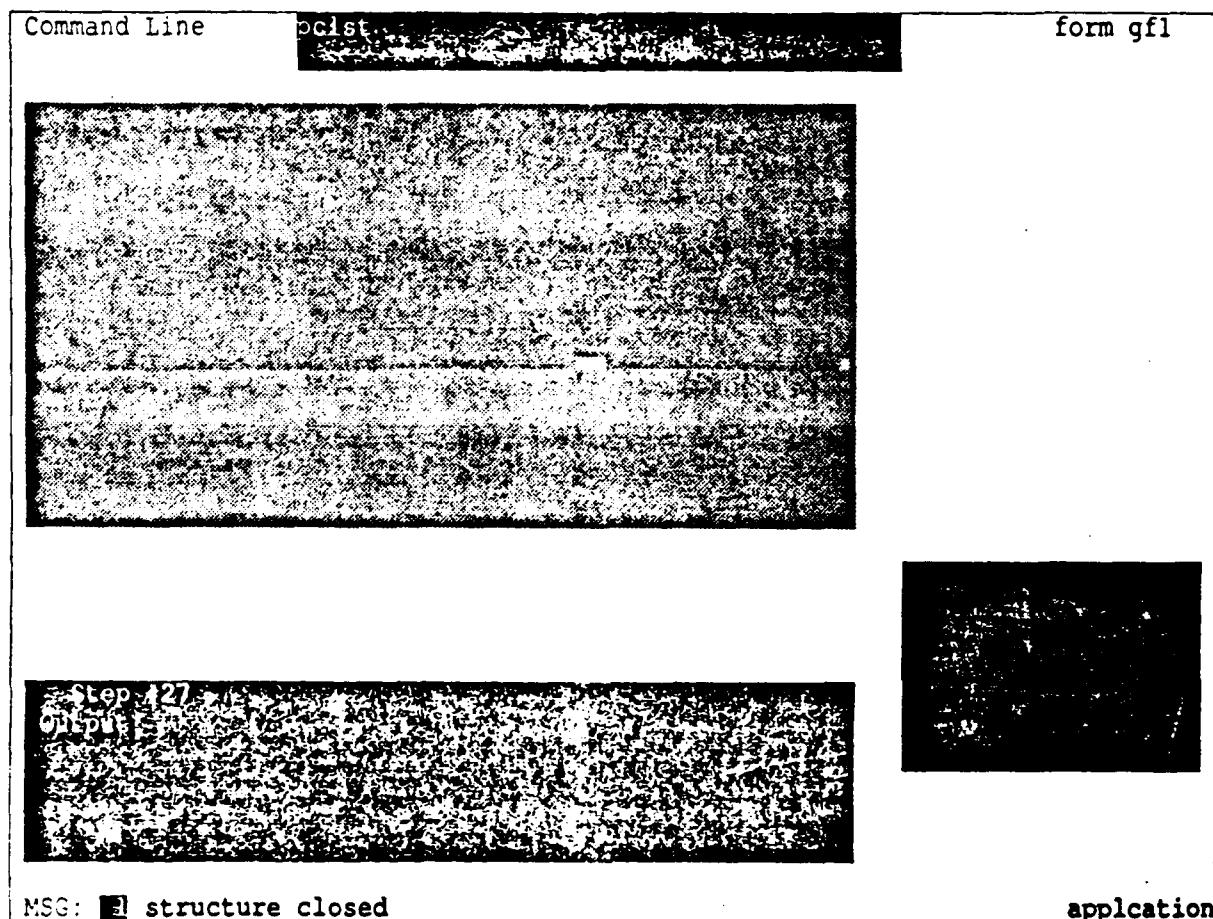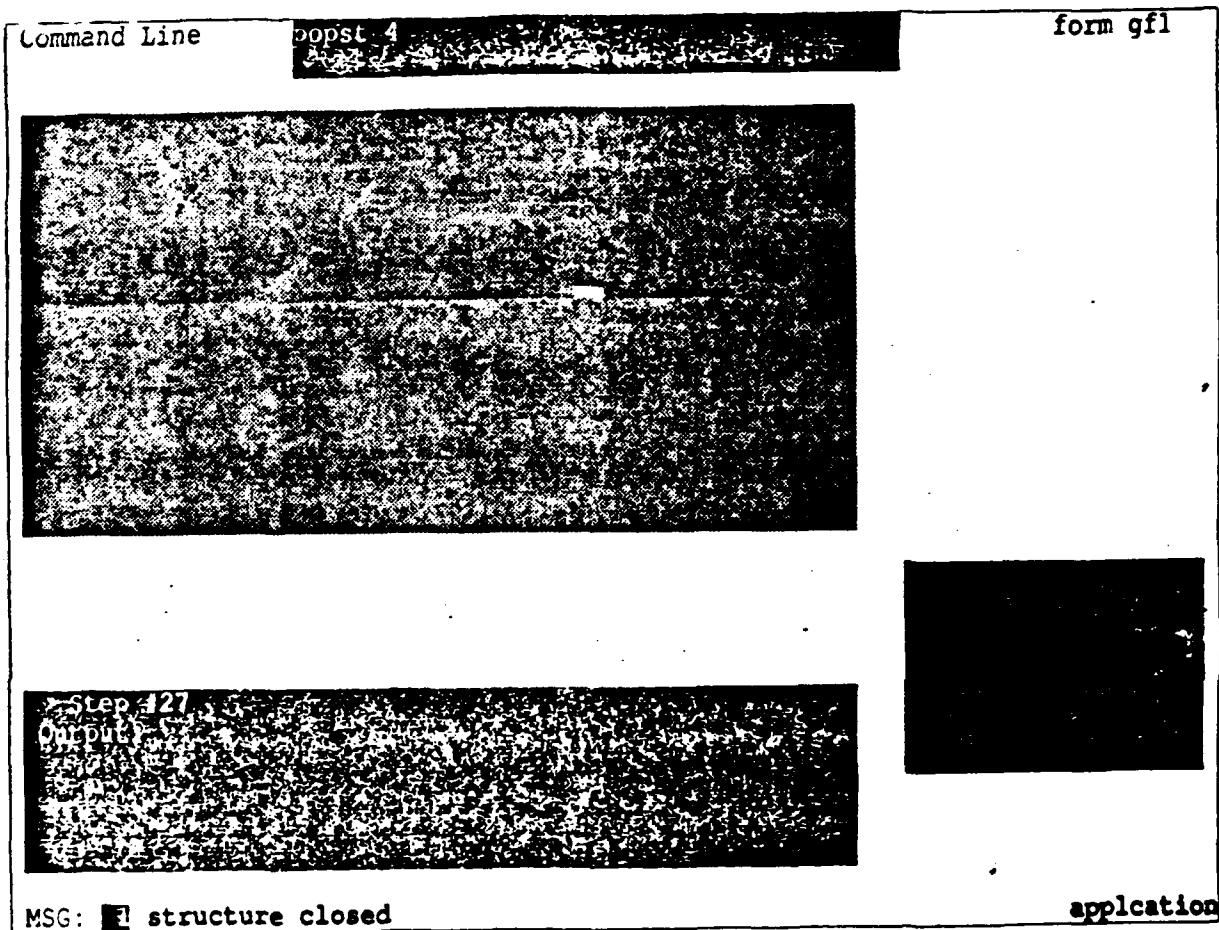
Figure 5-28b   (AFTER)

Figure 5-29a   (BEFORE)
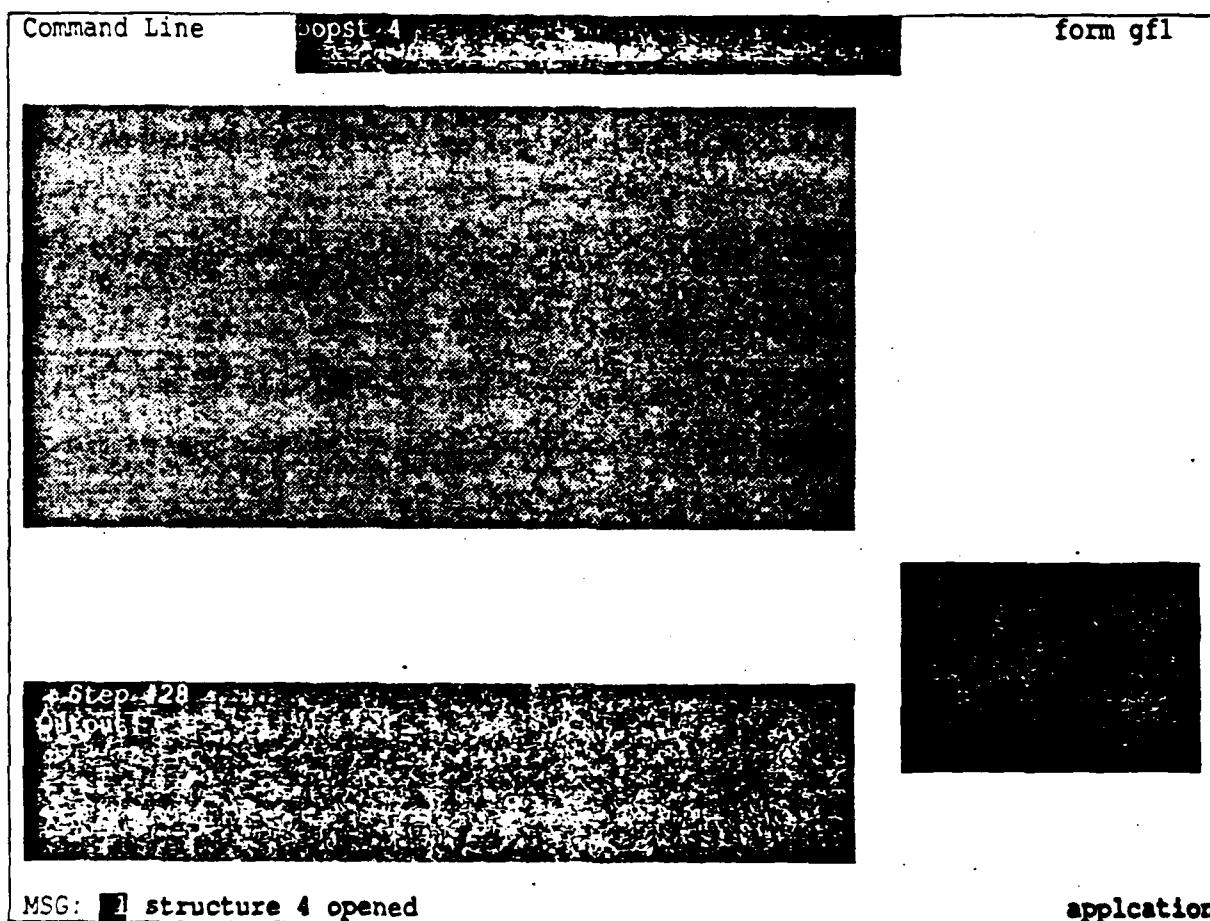
Figure 5-29b   (AFTER)
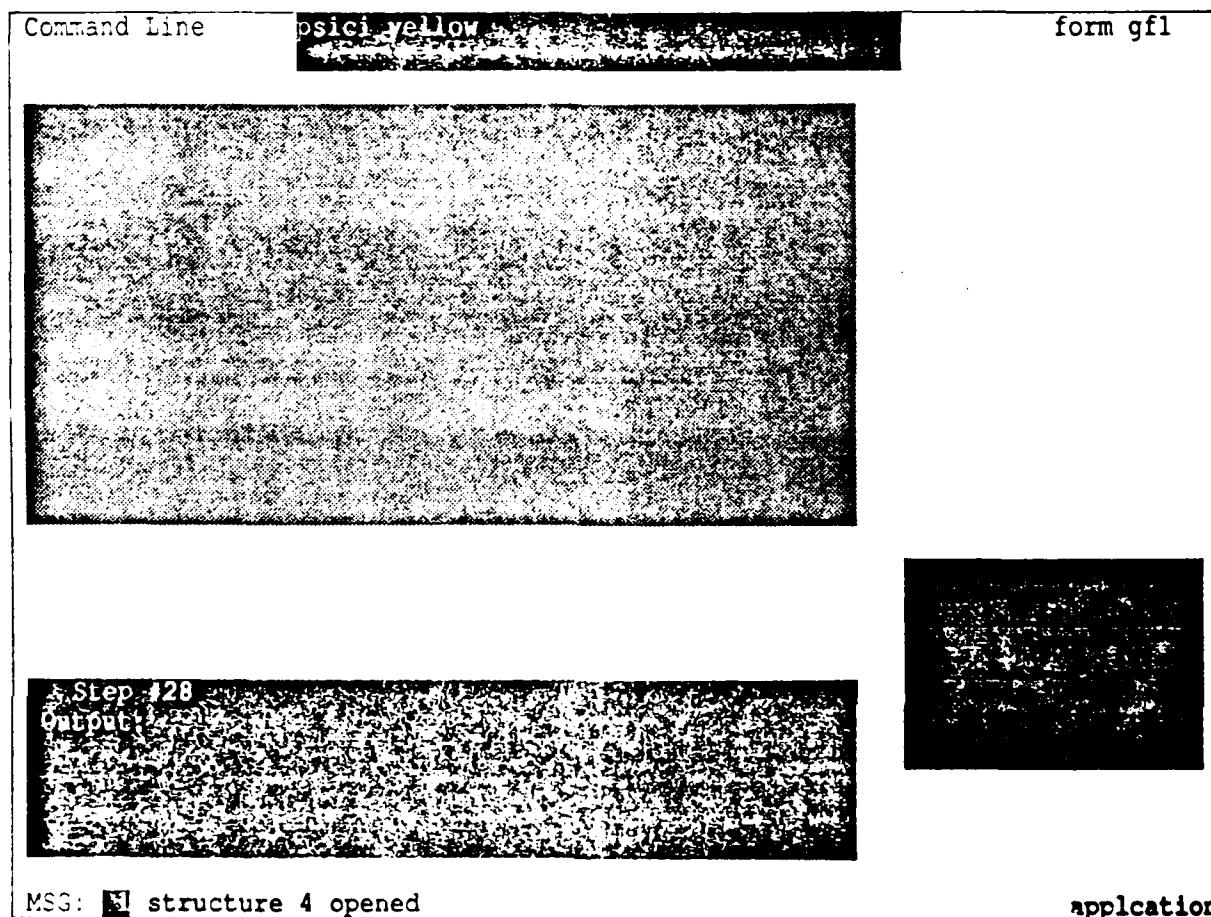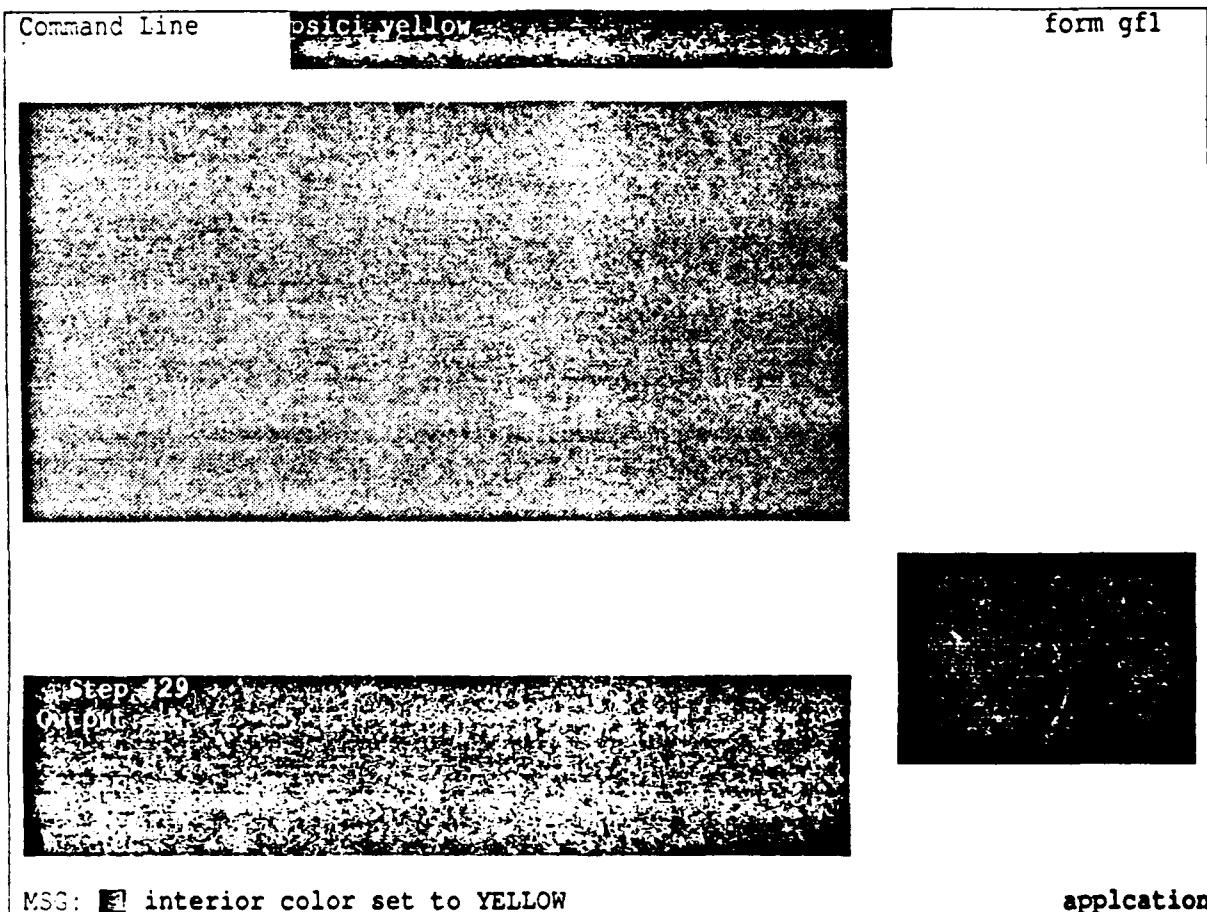
Command Line   psici yellow   form gfl

Step #28
Output
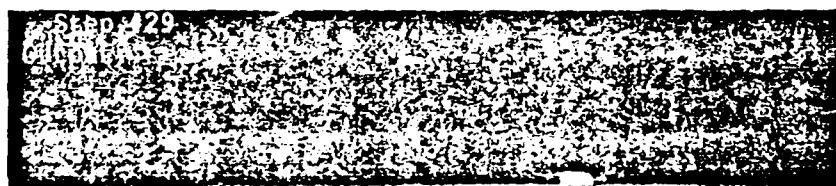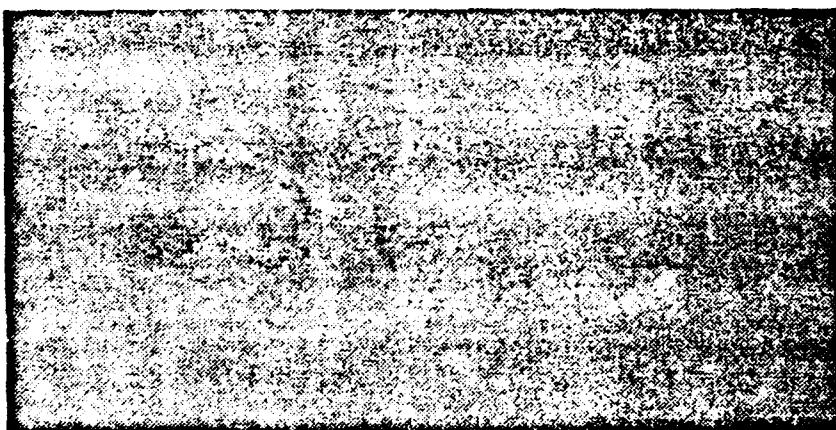
MSG: structure 4 opened   application

Figure 5-30a   (BEFORE)

Figure 5-30b (AFTER)

Figure 5-31a   (BEFORE)

Figure 5-31b   (AFTER)
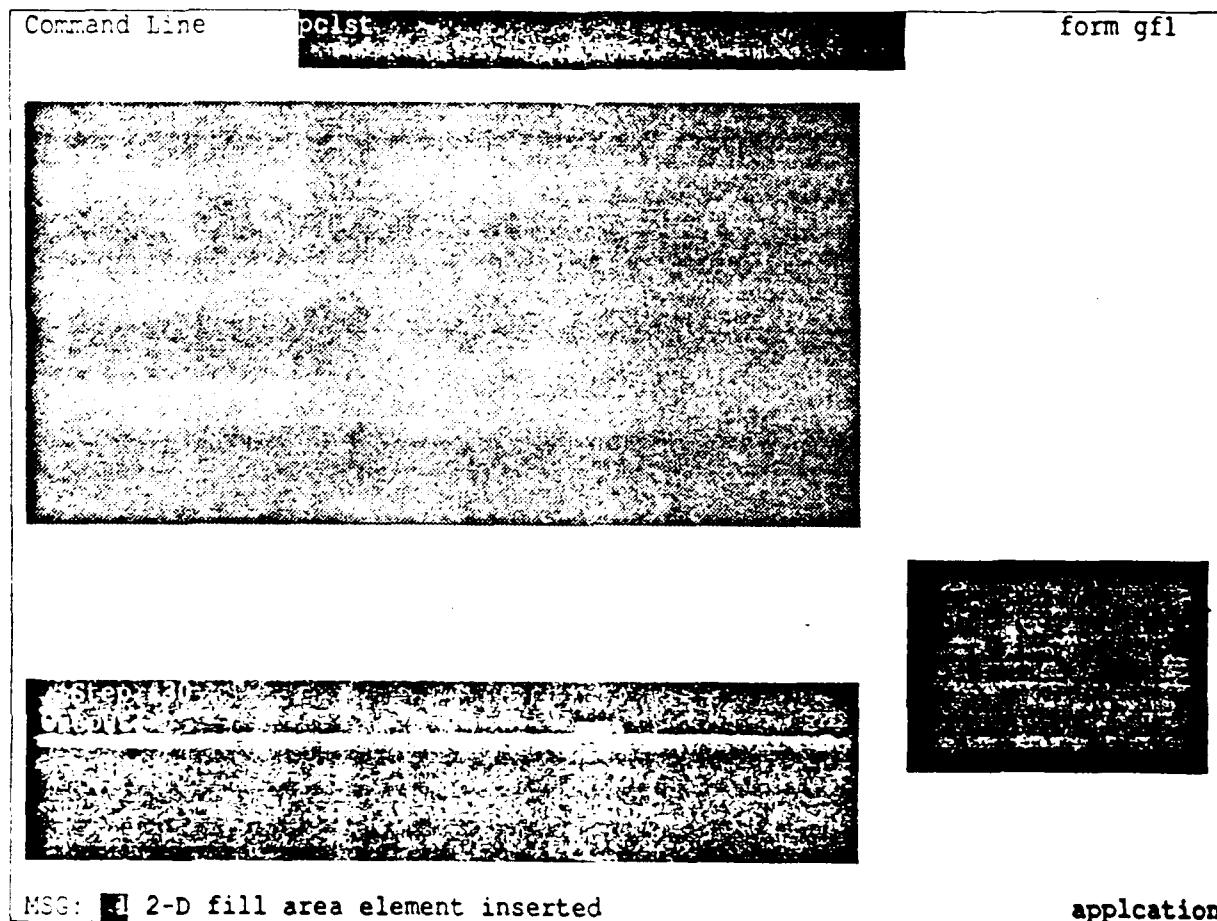
Command Line ......pclst...... form gfl

MSG: ▪ 2-D fill area element inserted .... applcation

Figure 5-32a (BEFORE)
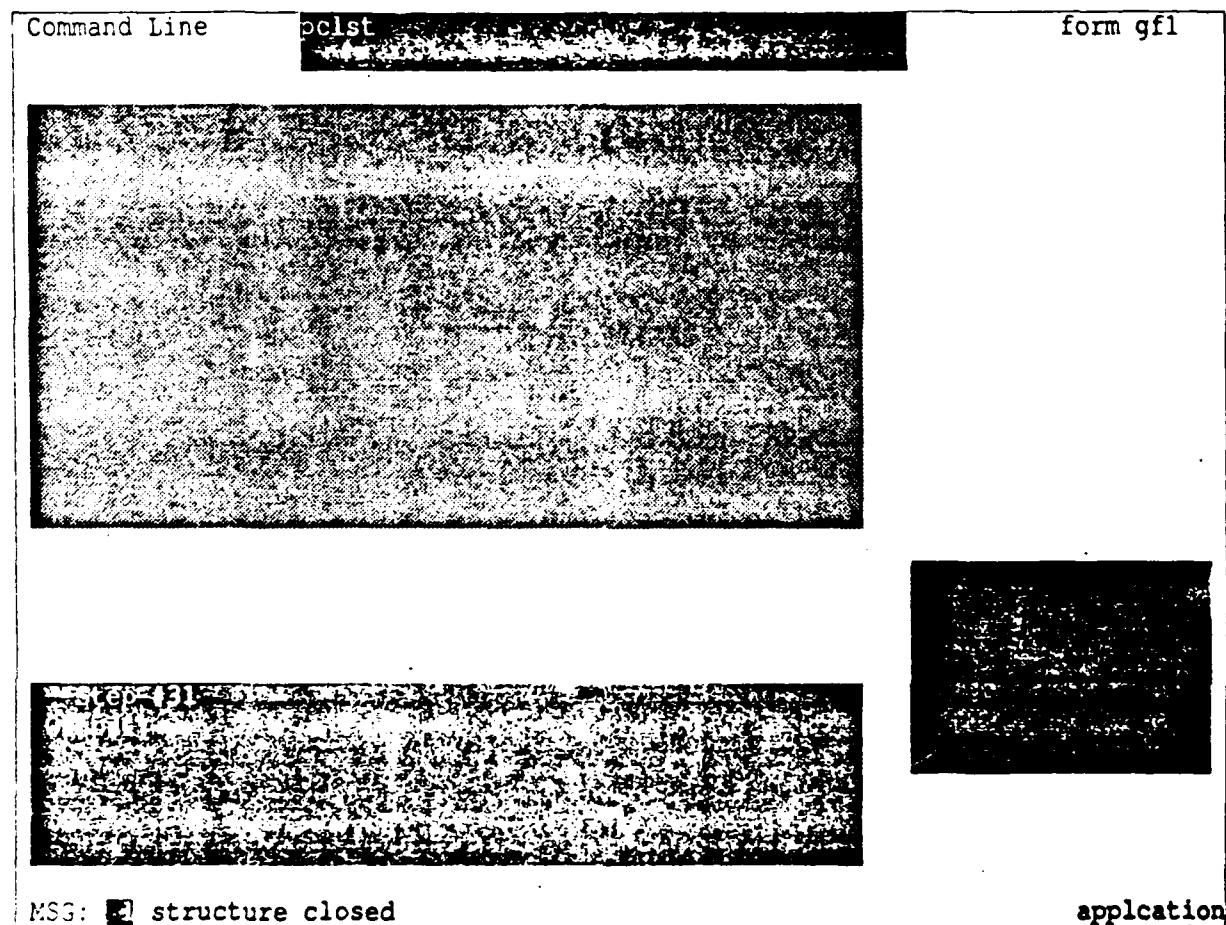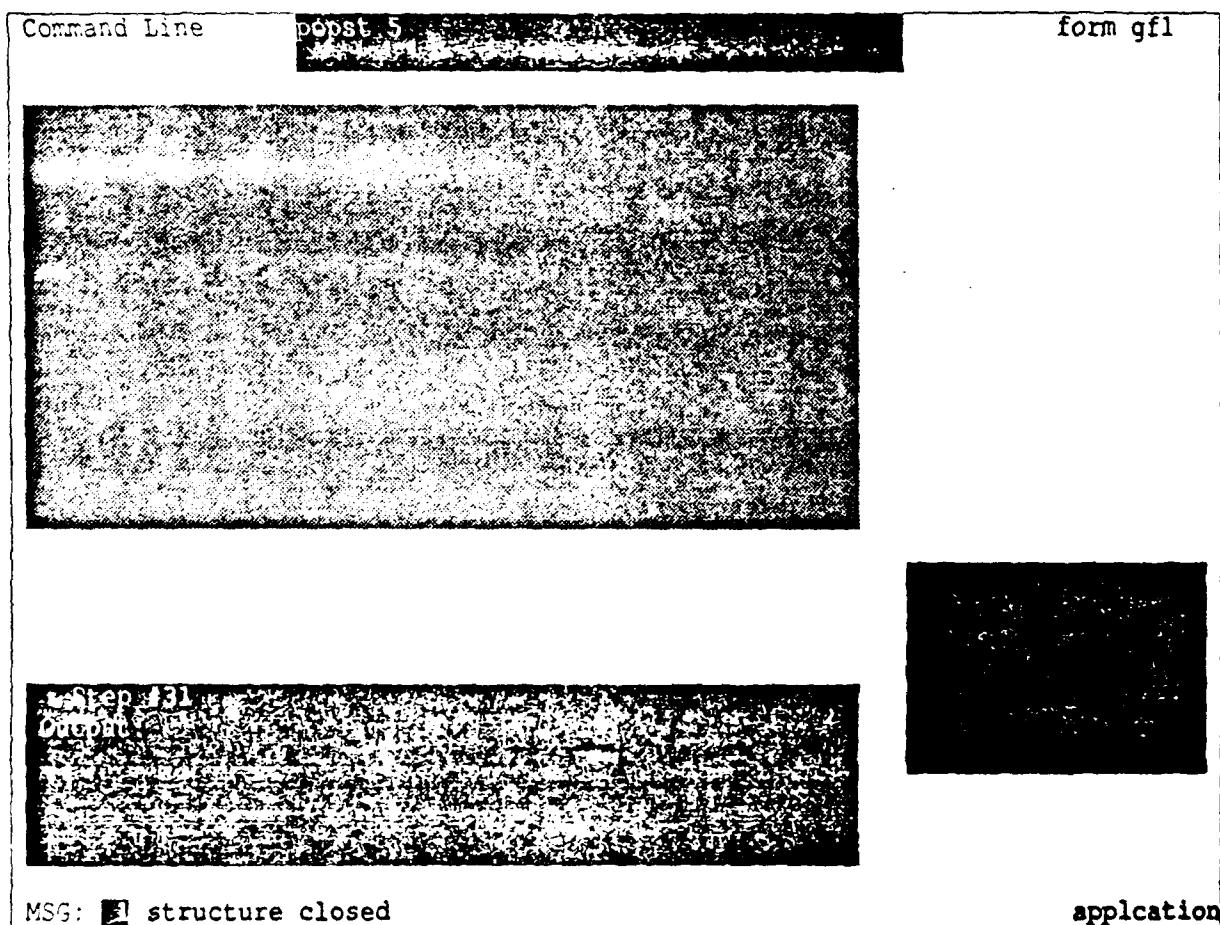
Figure 5-32b   (AFTER)

Figure 5-33a   (BEFORE)
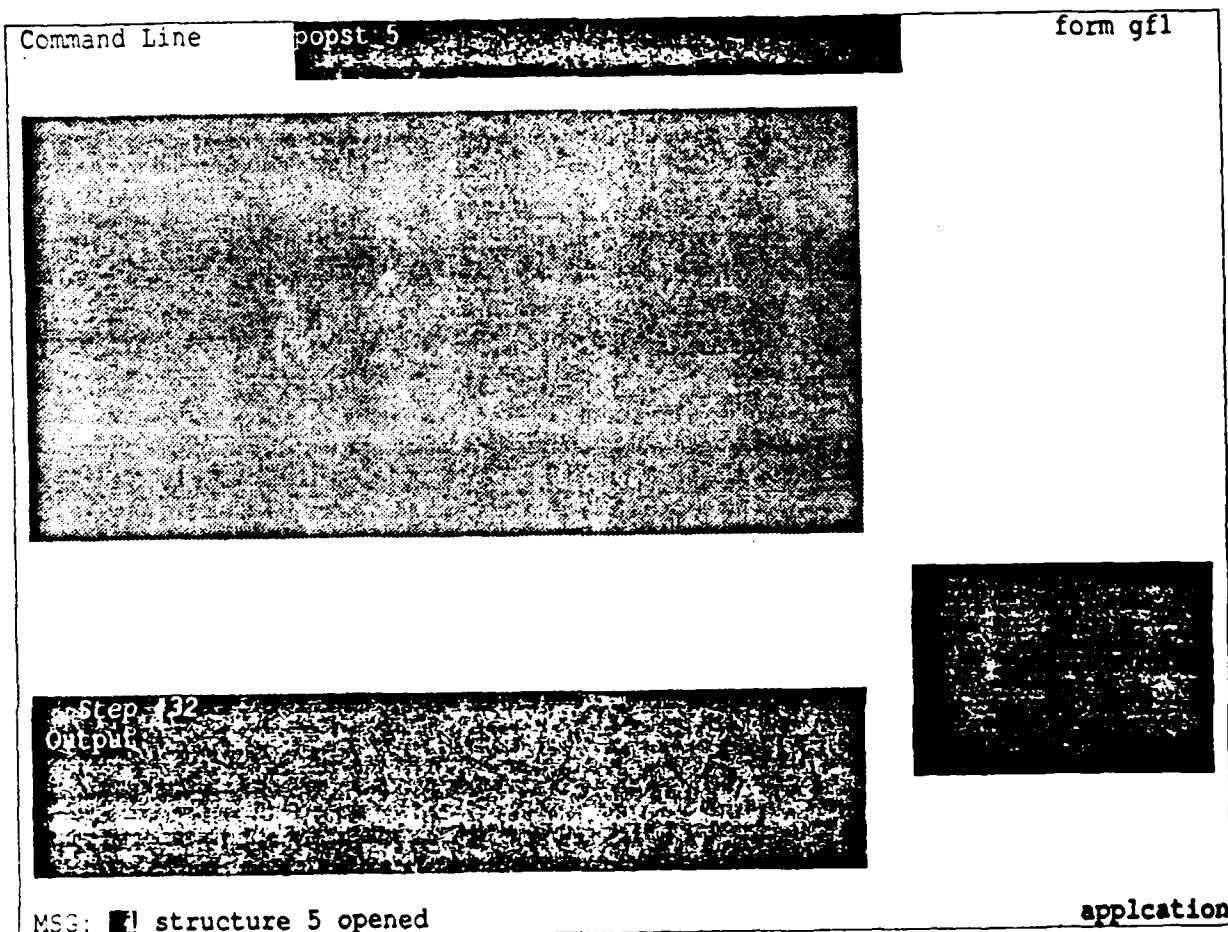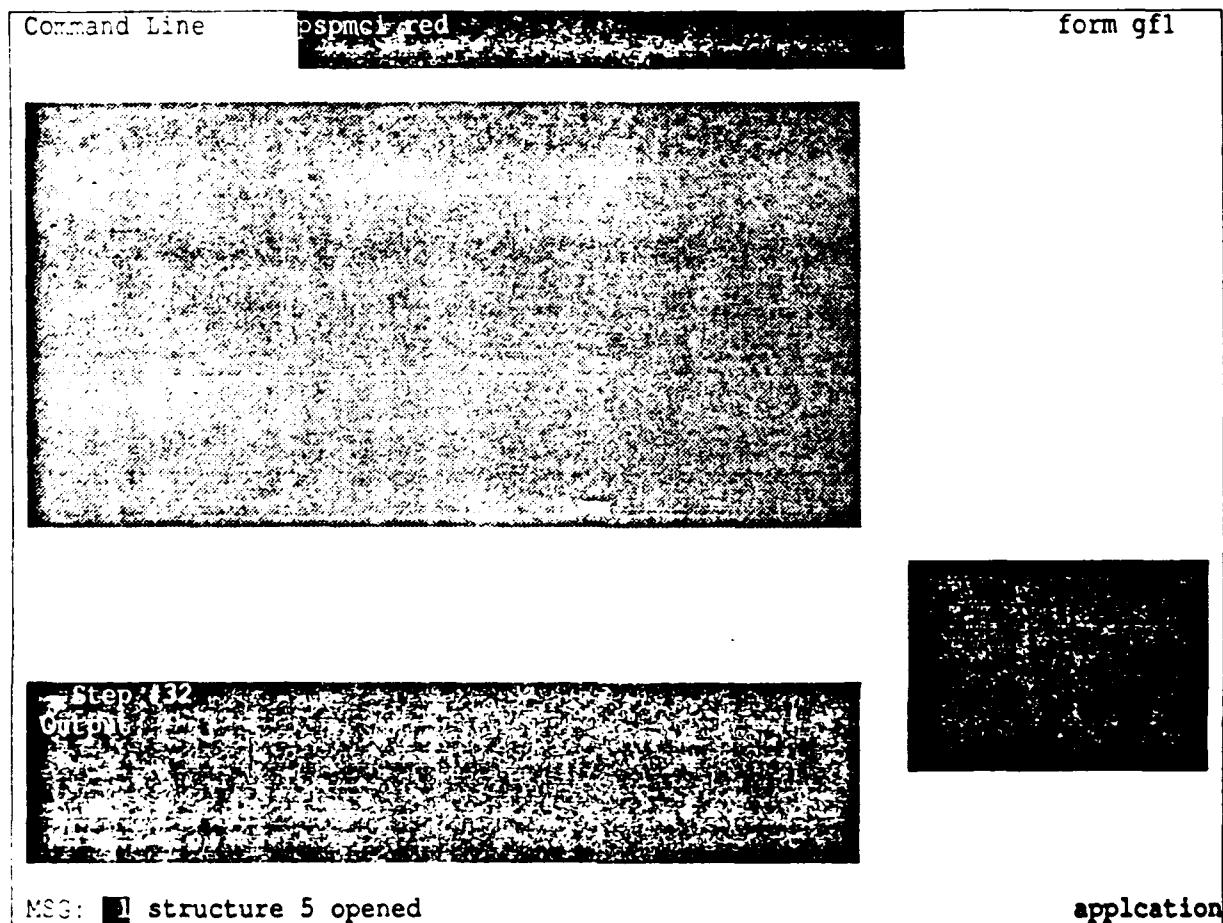
Figure 5-33b   (AFTER)

Figure 5-34a   (BEFORE)
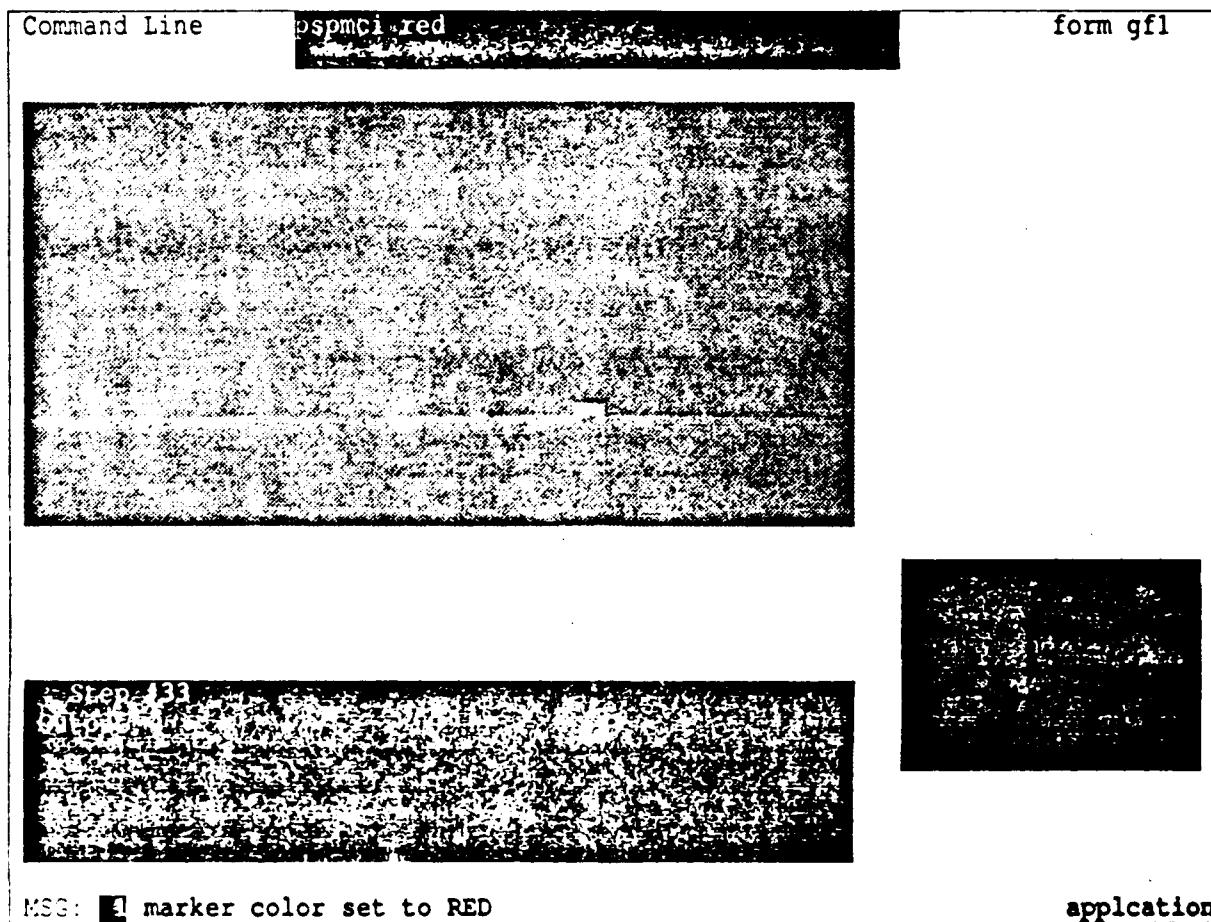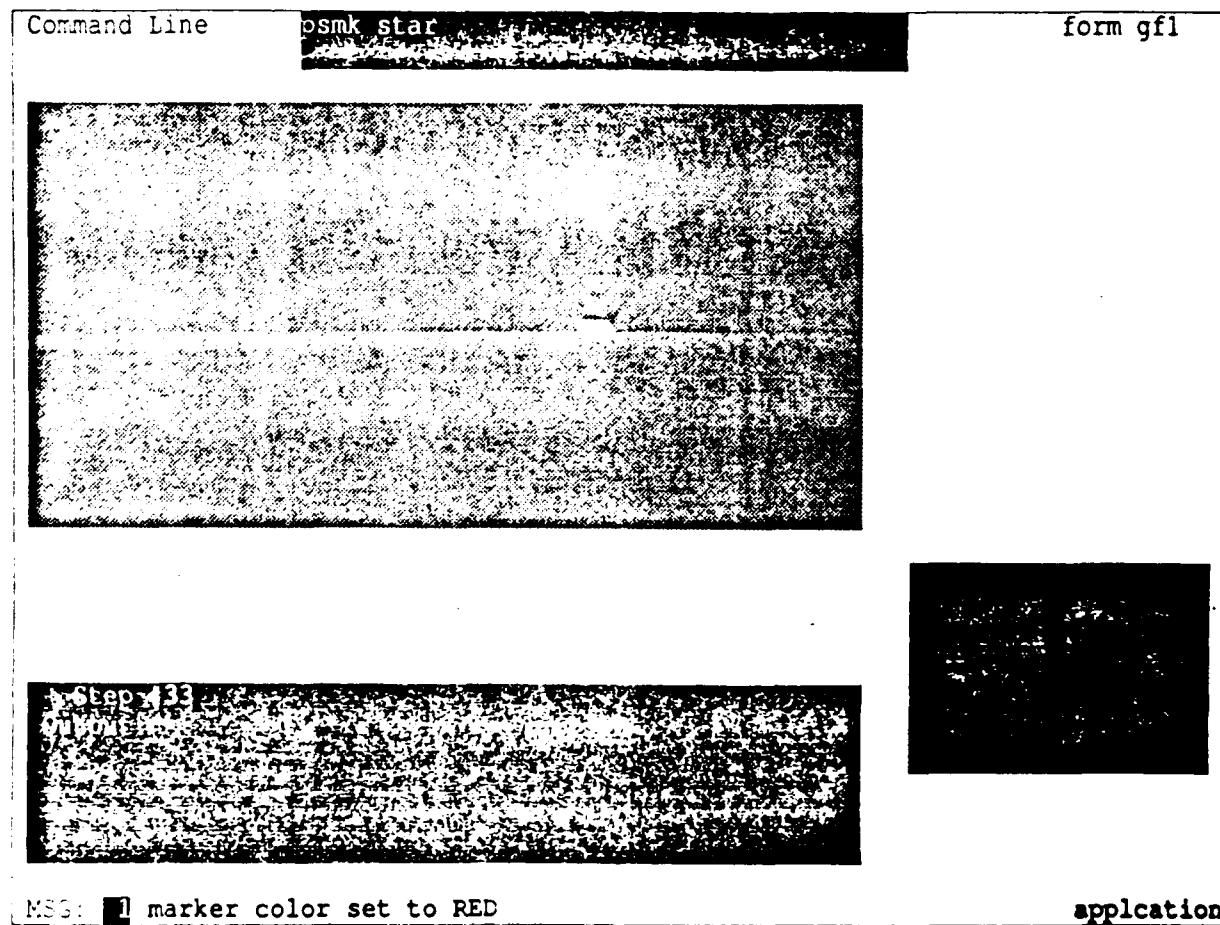
Figure 5-34b   (AFTER)

Figure 5-35a   (BEFORE)
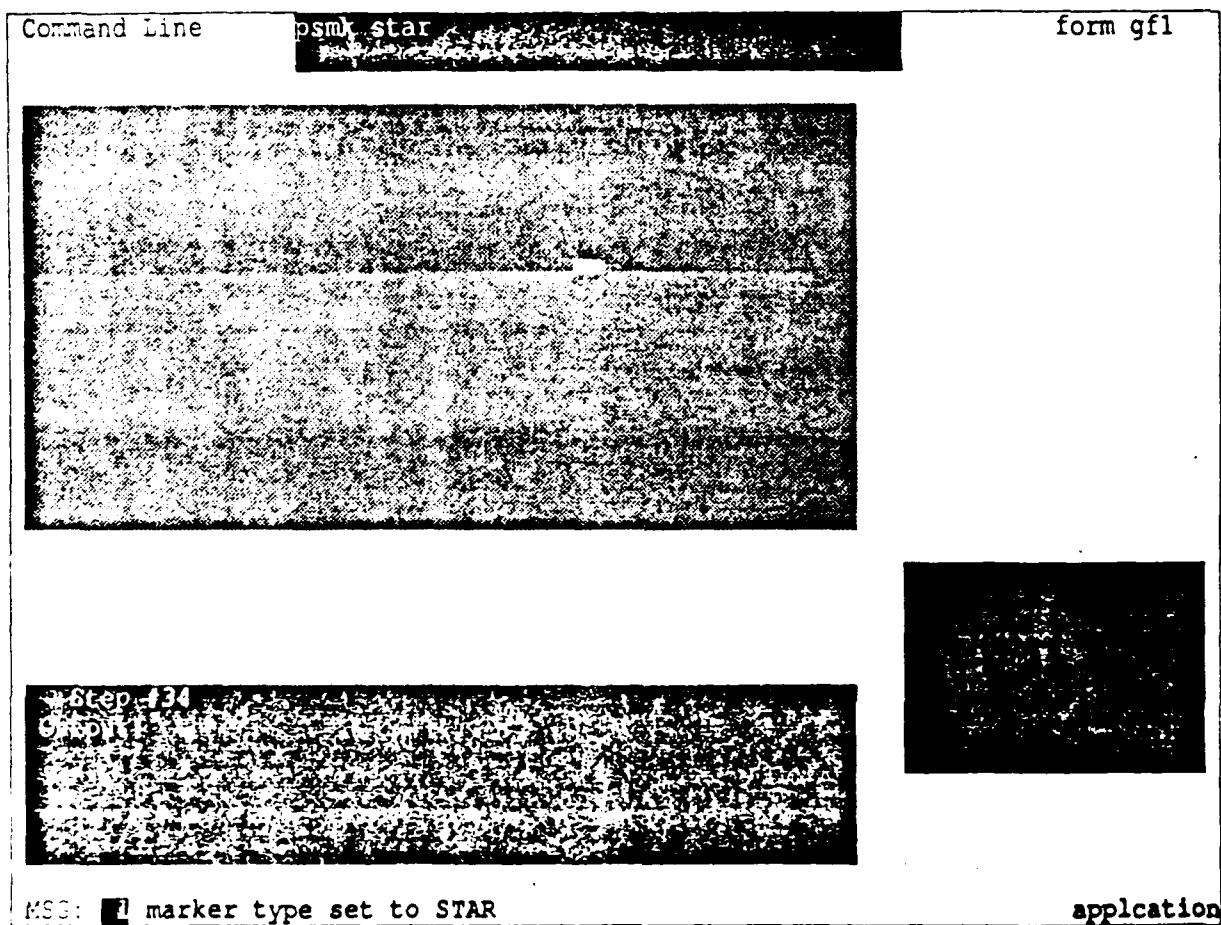
Figure 5-35b   (AFTER)
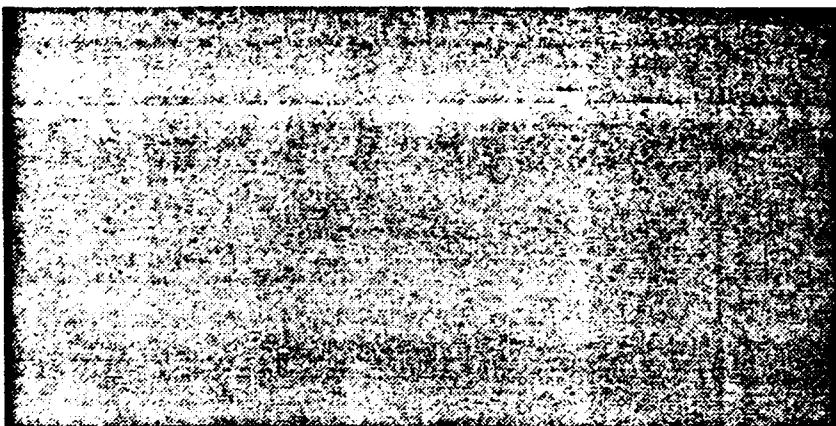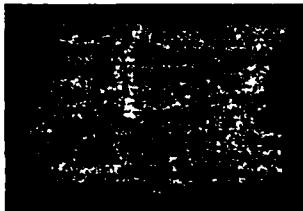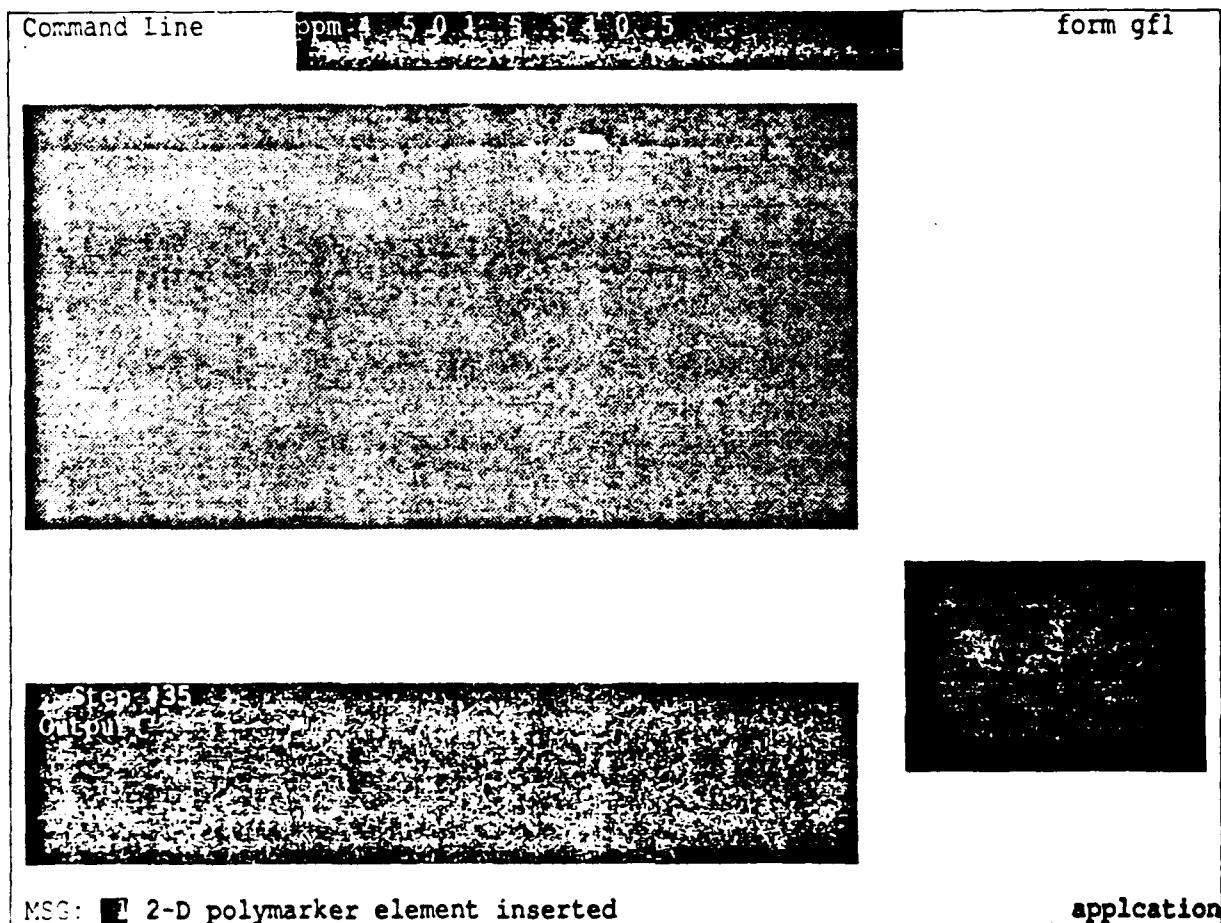
Figure 5-36a   (BEFORE)

Figure 5-36b   (AFTER)
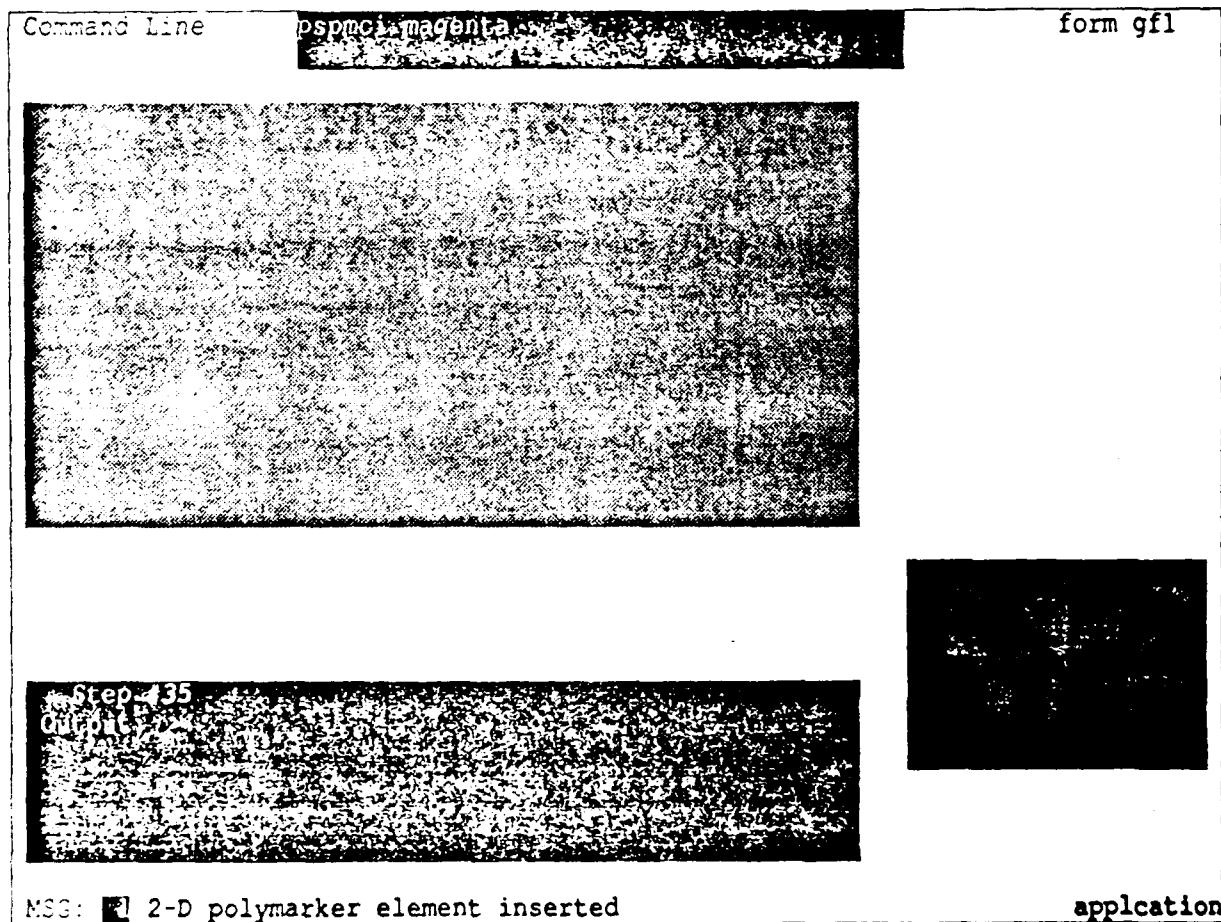
Figure 5-37a   (BEFORE)

Figure 5-37b   (AFTER)

Command Line   qsmk circle   form 7.1

Step #36

MSG: ⬛1 marker color set to MAGENTA   applcation
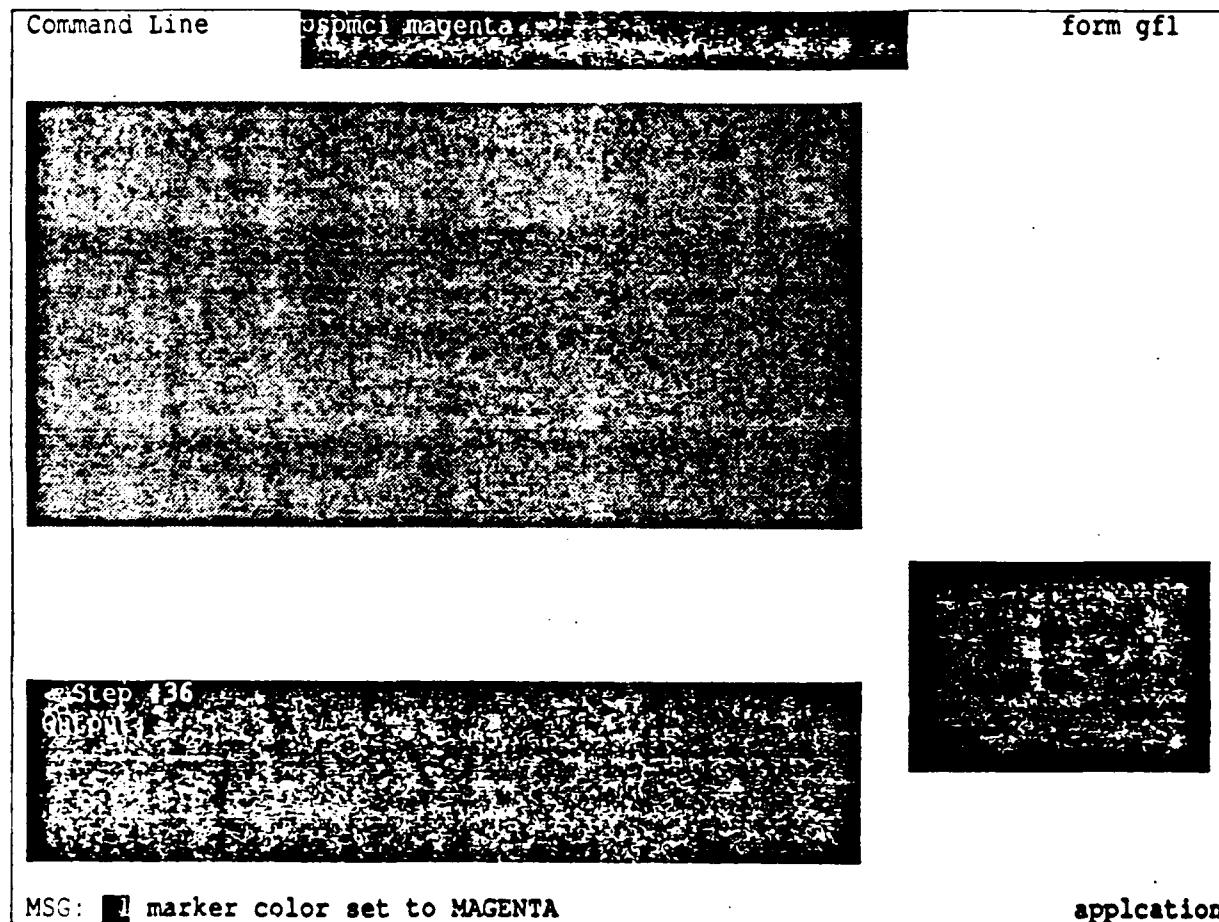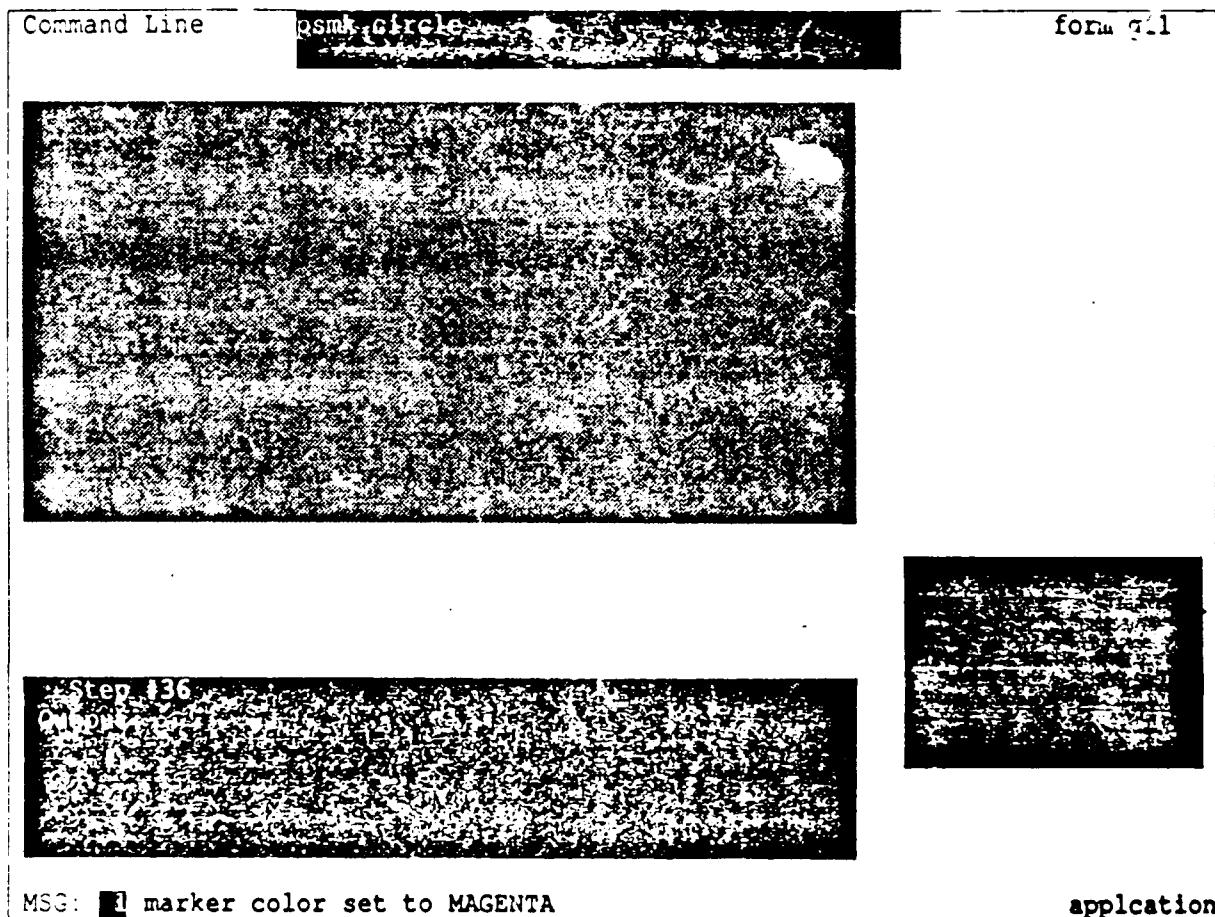
Figure 5-38a   (BEFORE)
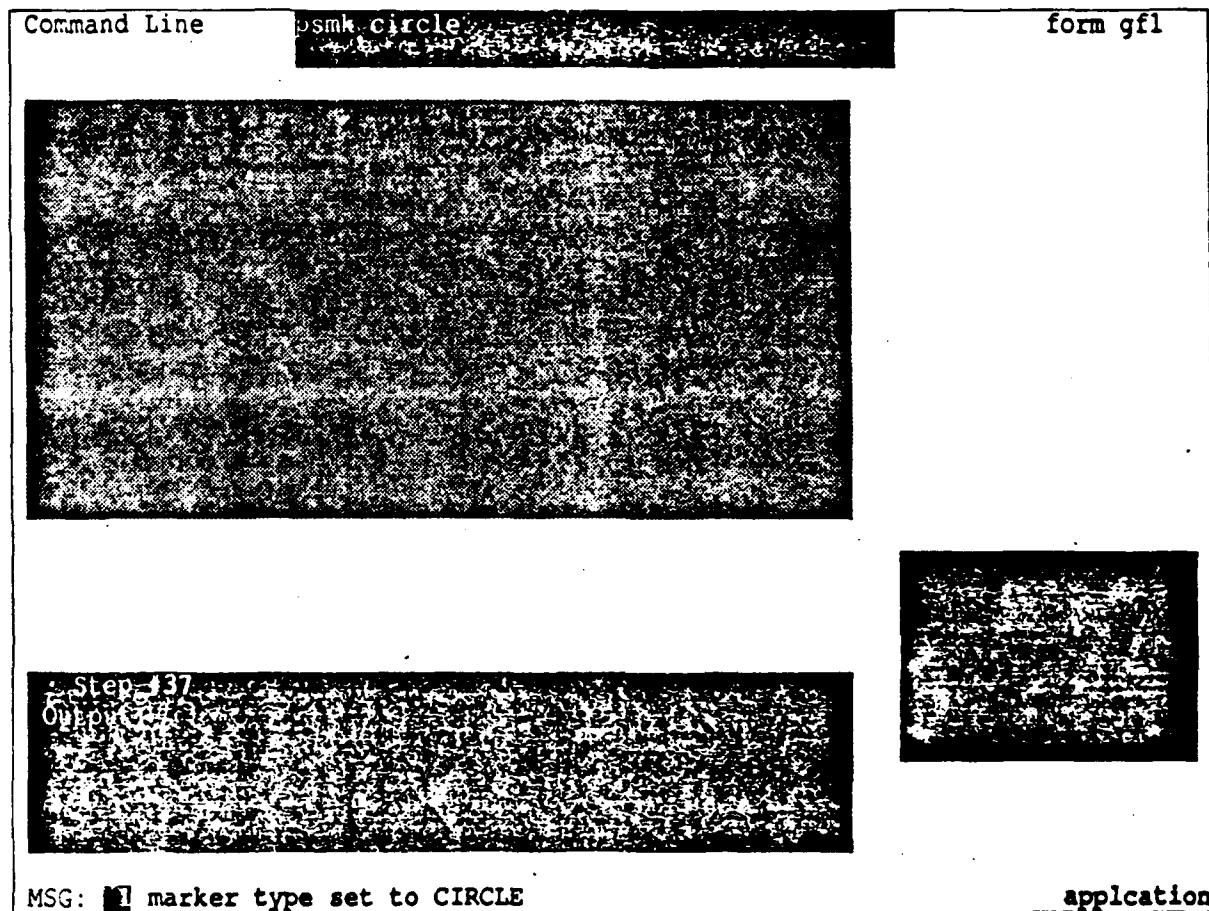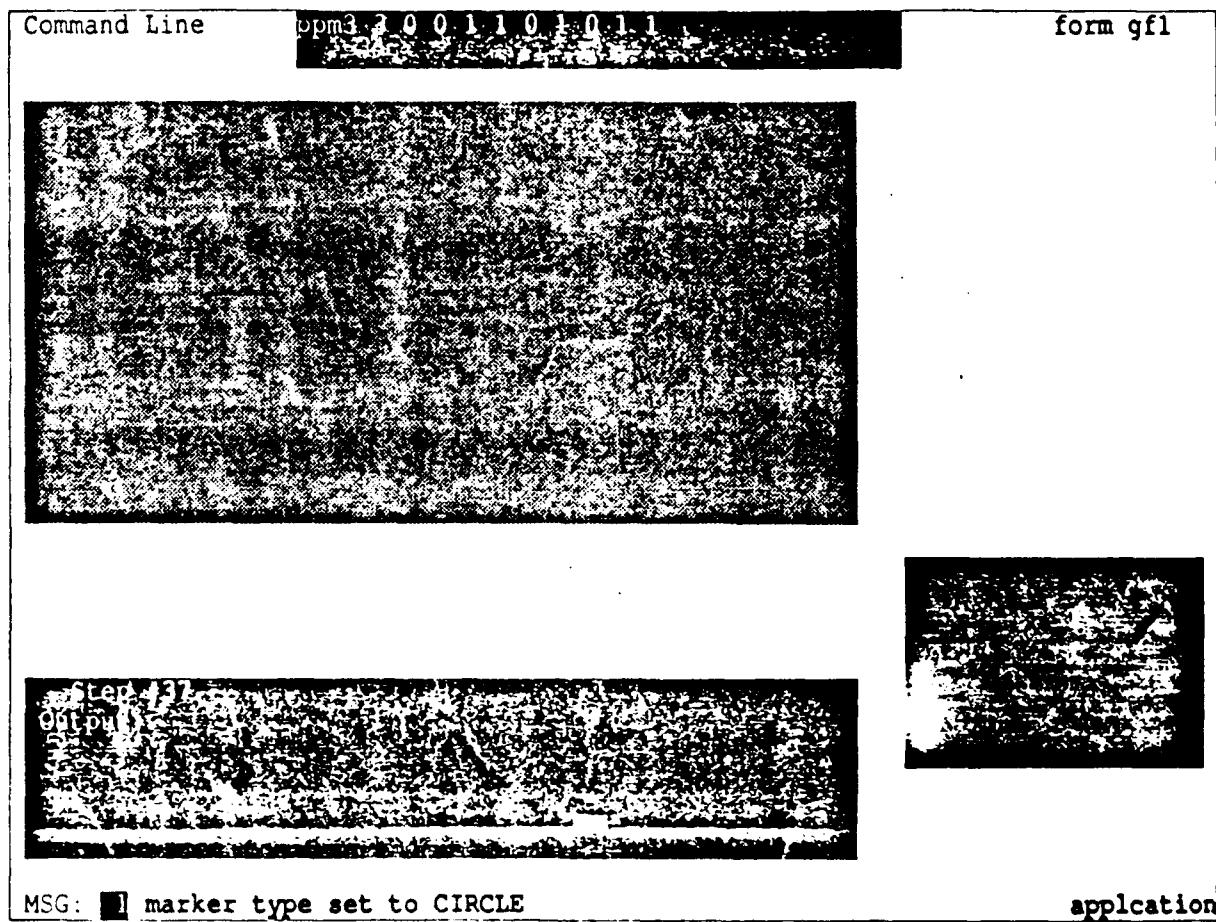
Figure 5-38b   (AFTER)

Figure 5-39a   (BEFORE)
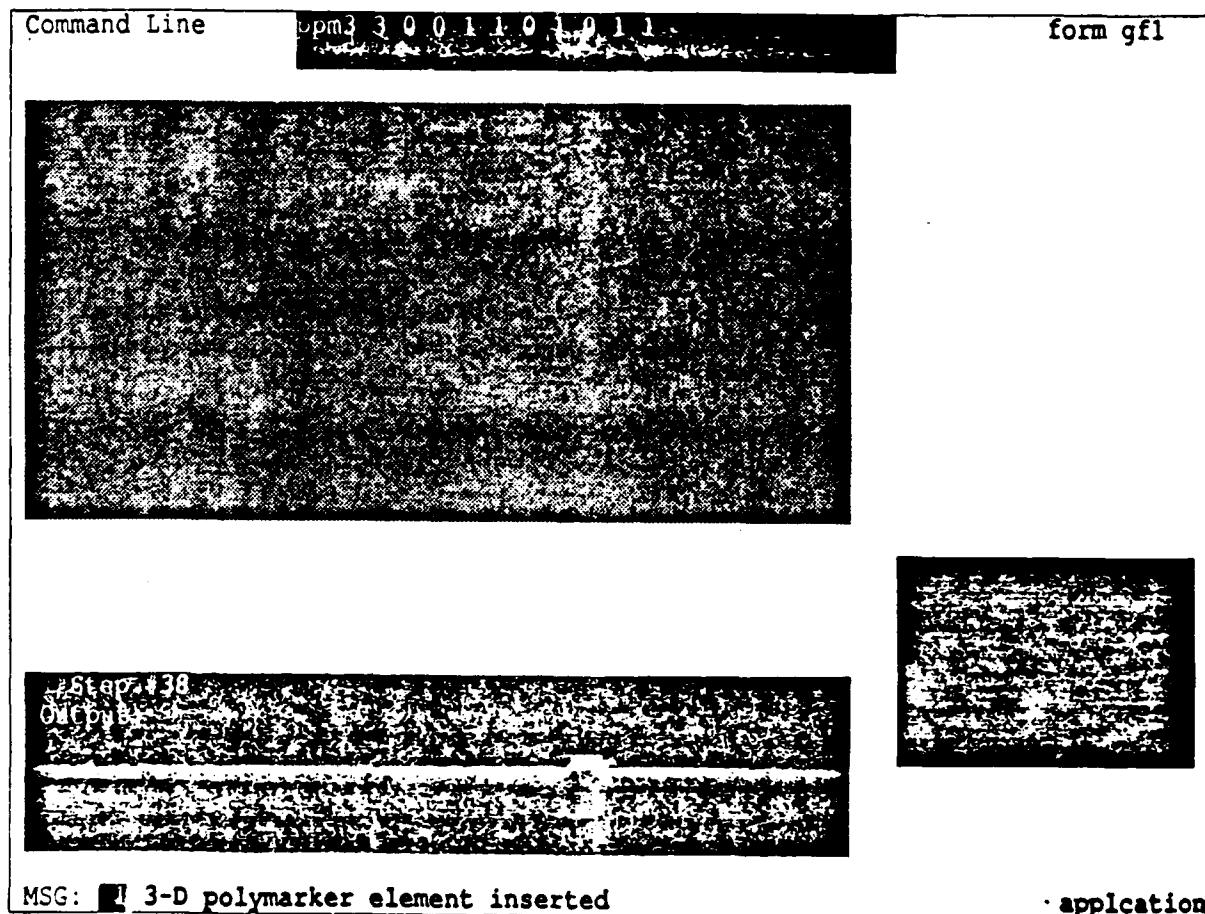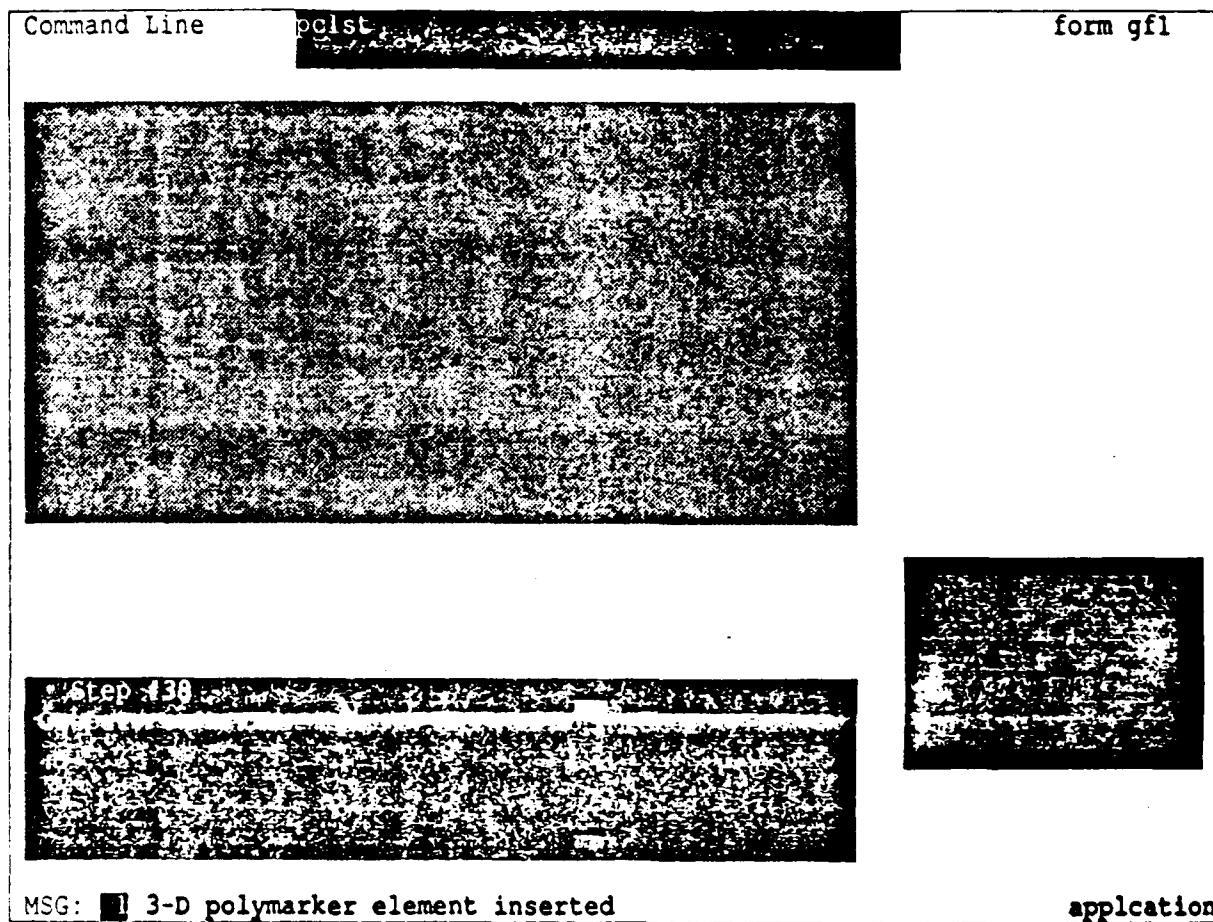
Figure 5-39b   (AFTER)

Figure 5-40a   (BEFORE)
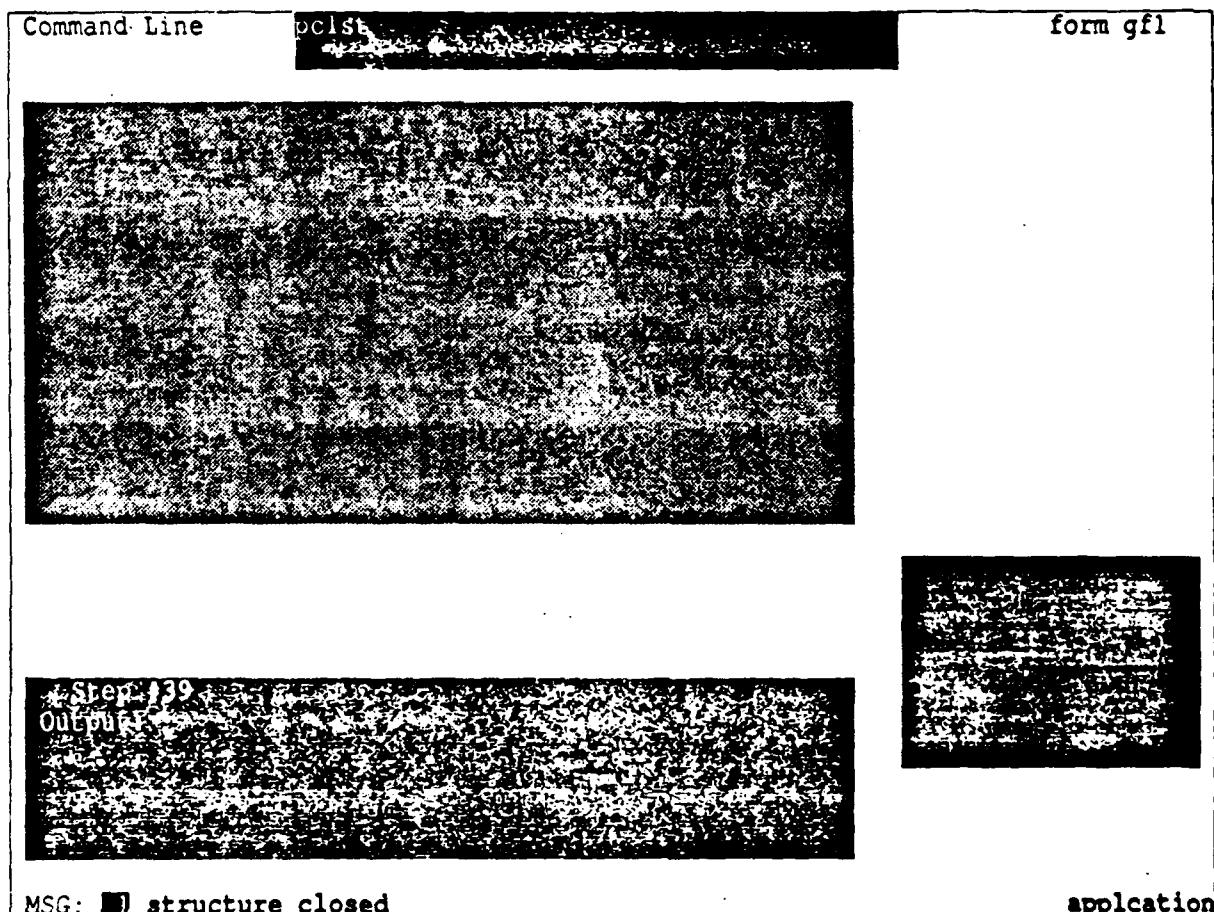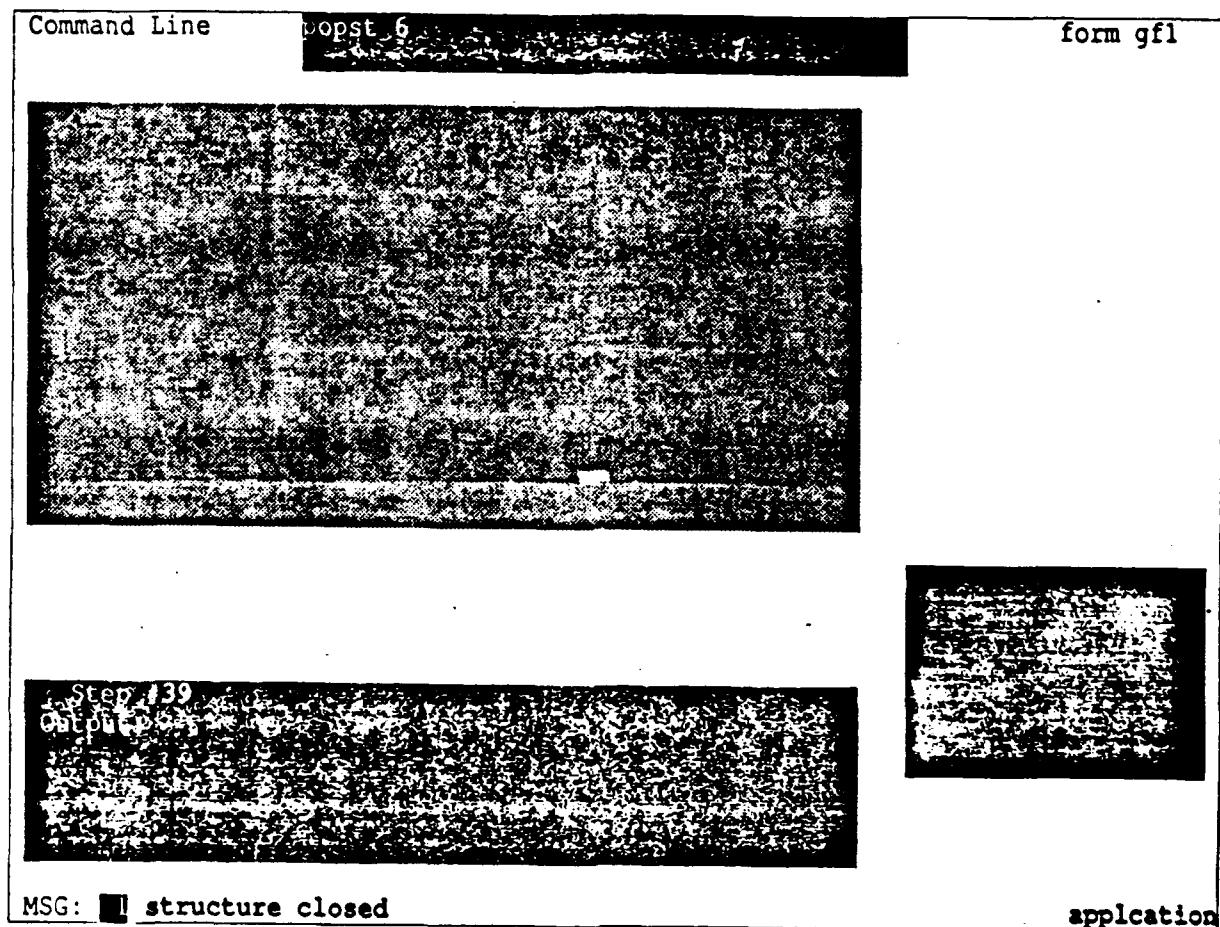
Figure 5-40b   (AFTER)

Figure 5-41a   (BEFORE)

Command Line   popst 6   form gfl

Step 140
Output

MSG: ▉ structure 6 opened   application
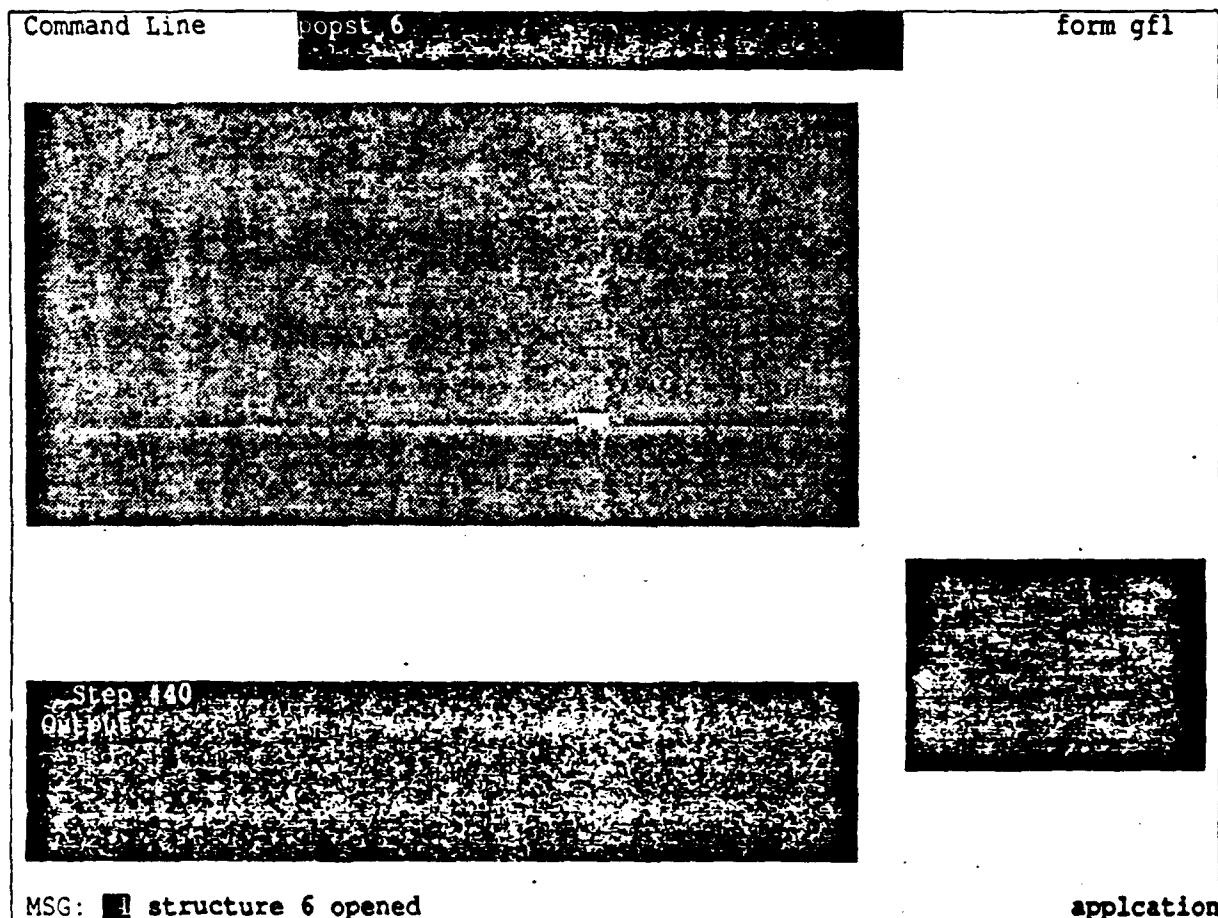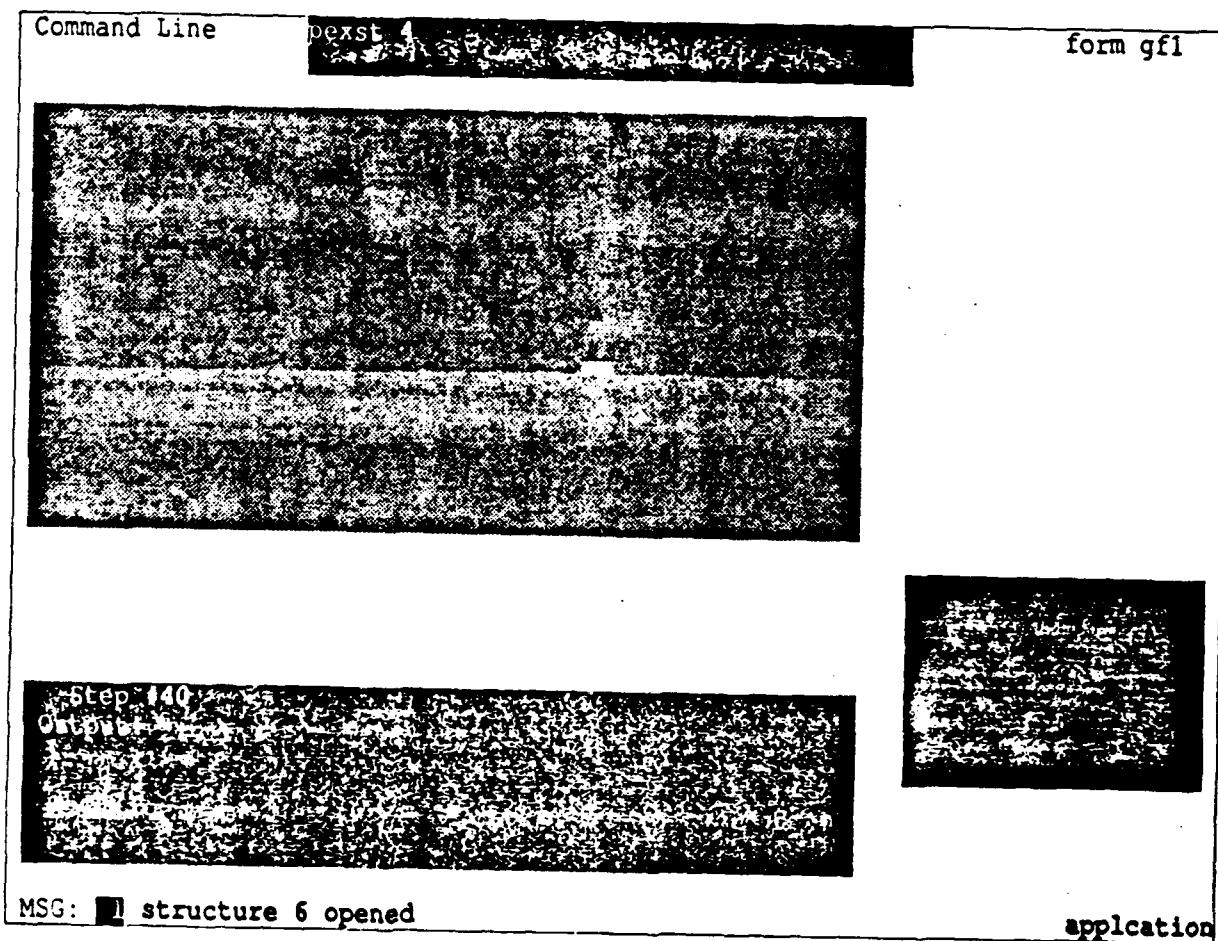
Figure 5-41b   (AFTER)

Figure 5-42a   (BEFORE)

Figure 5-42b   (AFTER)

Figure 5-43a   (BEFORE)

Command Line   pexst   form gfl

Step #42
Output

MSG: ▆ execute structure 3 element inserted   applcation
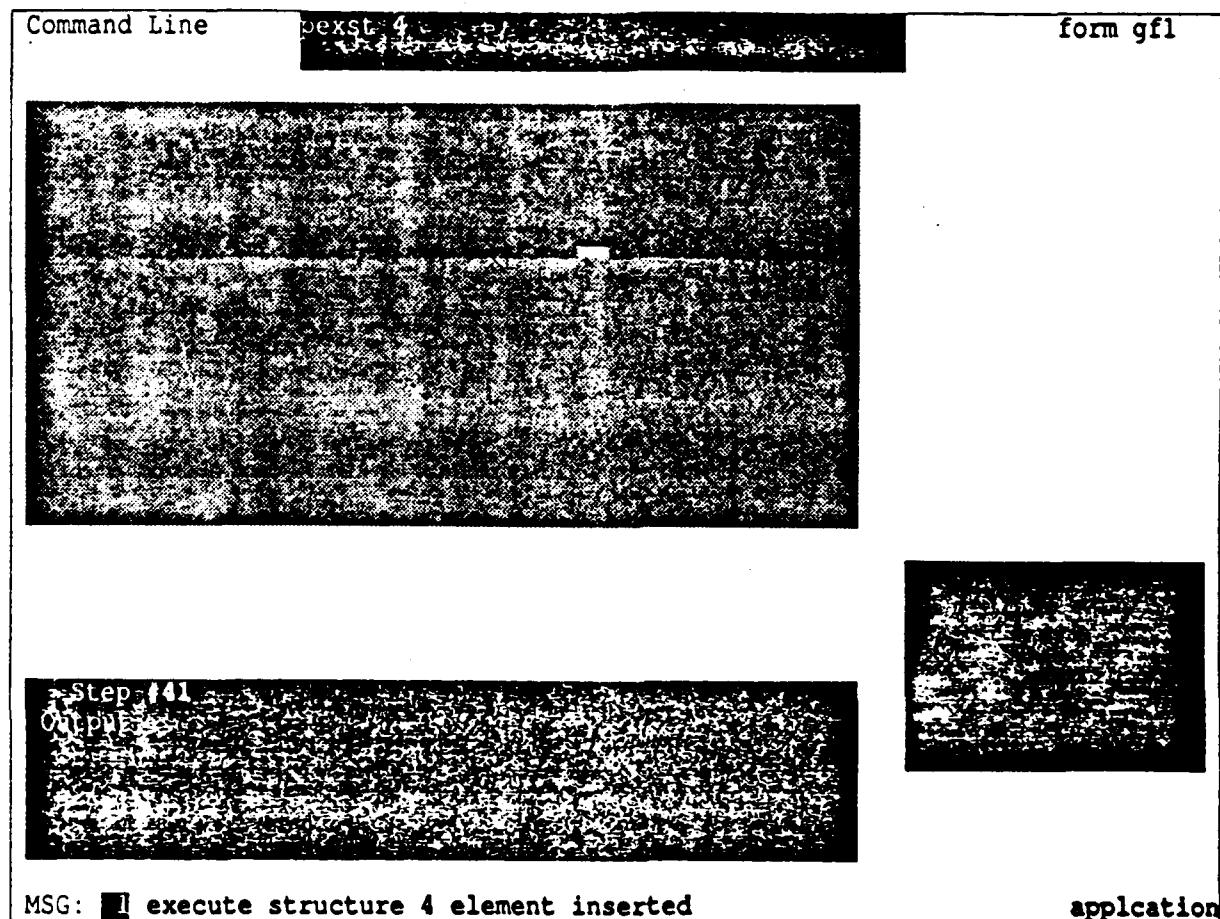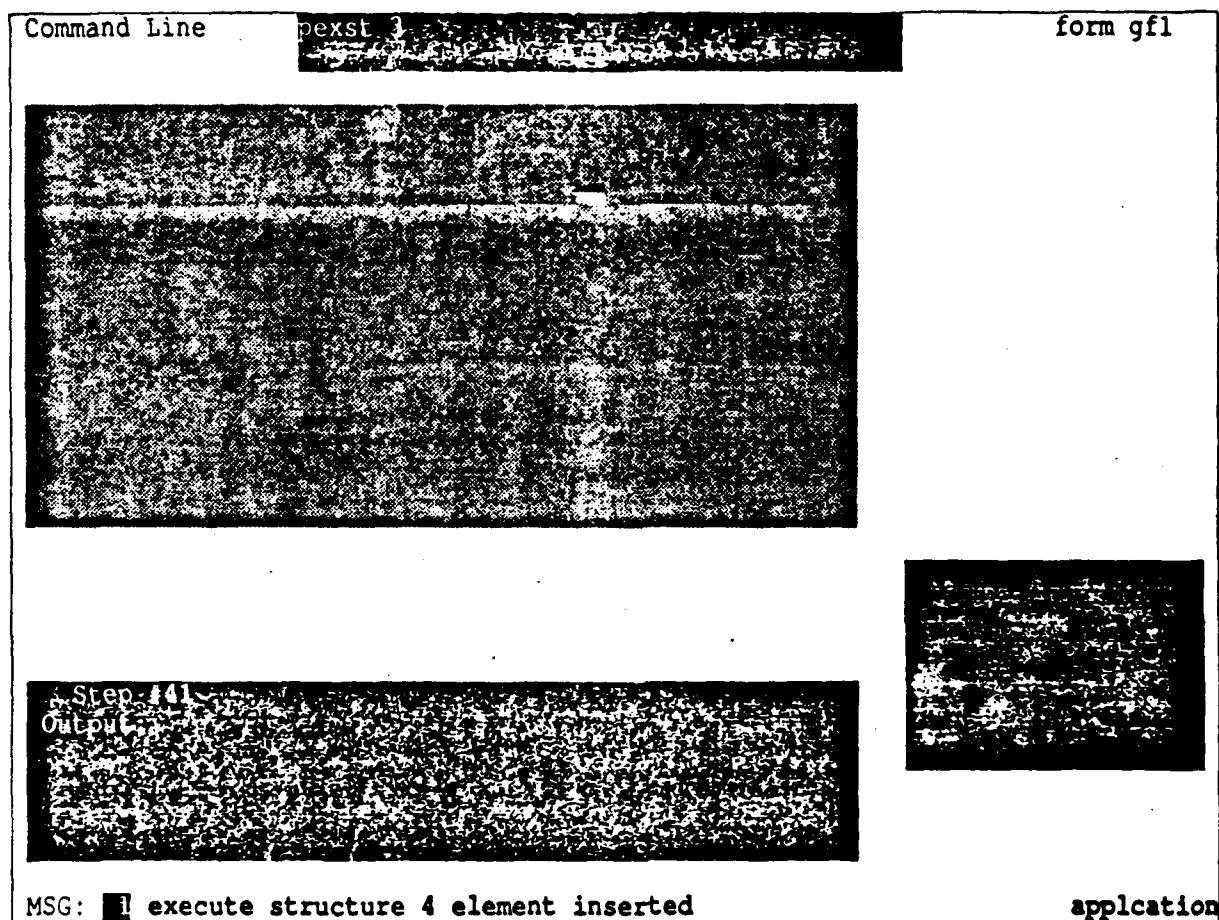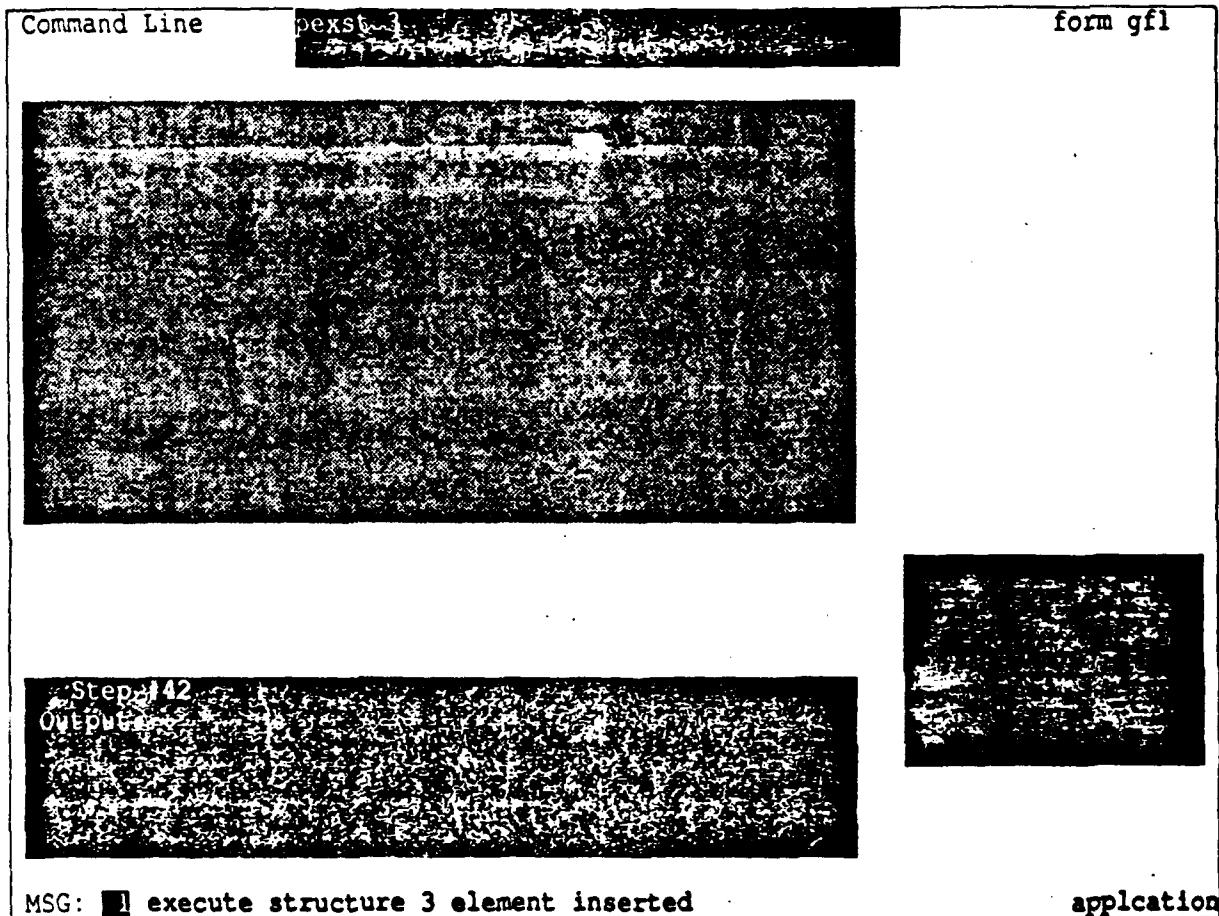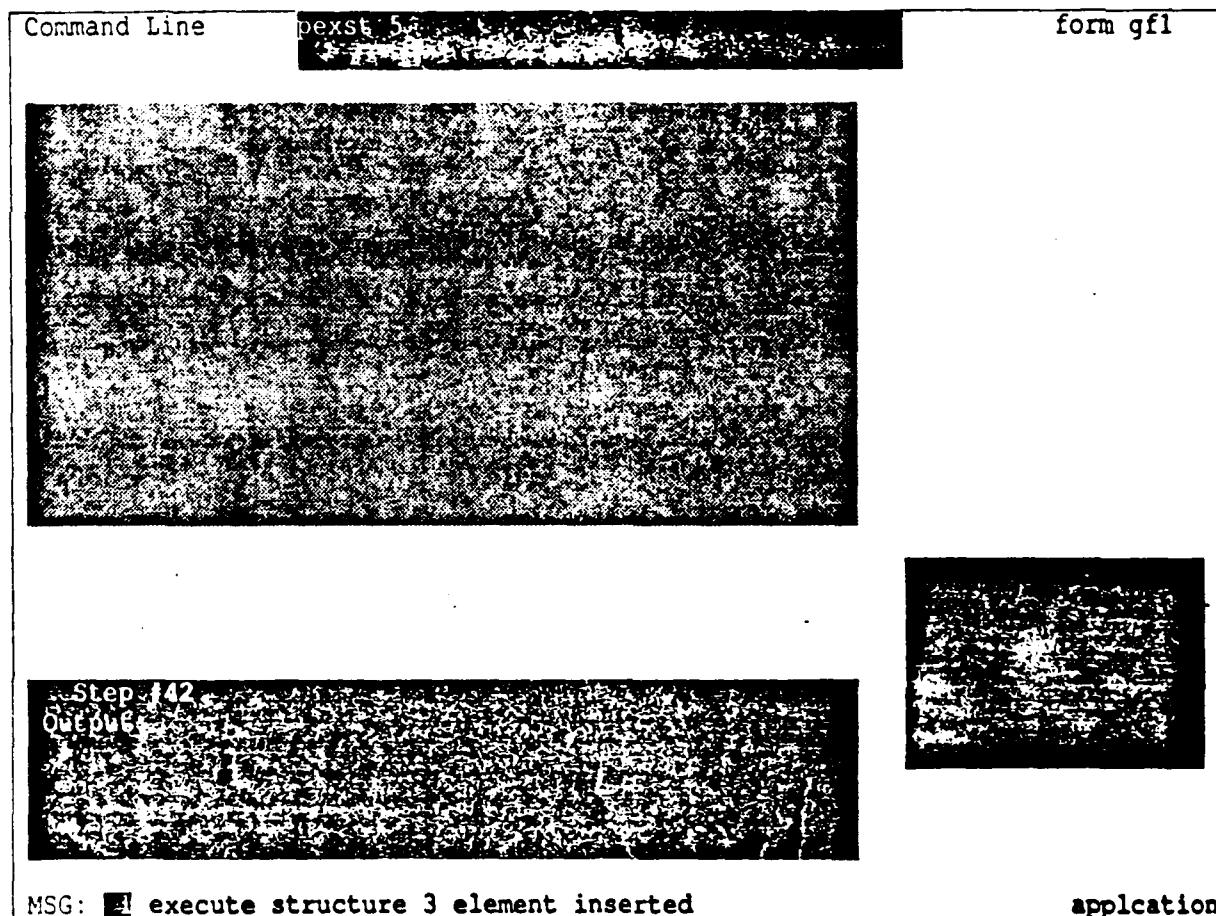
Figure 5-43b   (AFTER)

Figure 5-44a   (BEFORE)
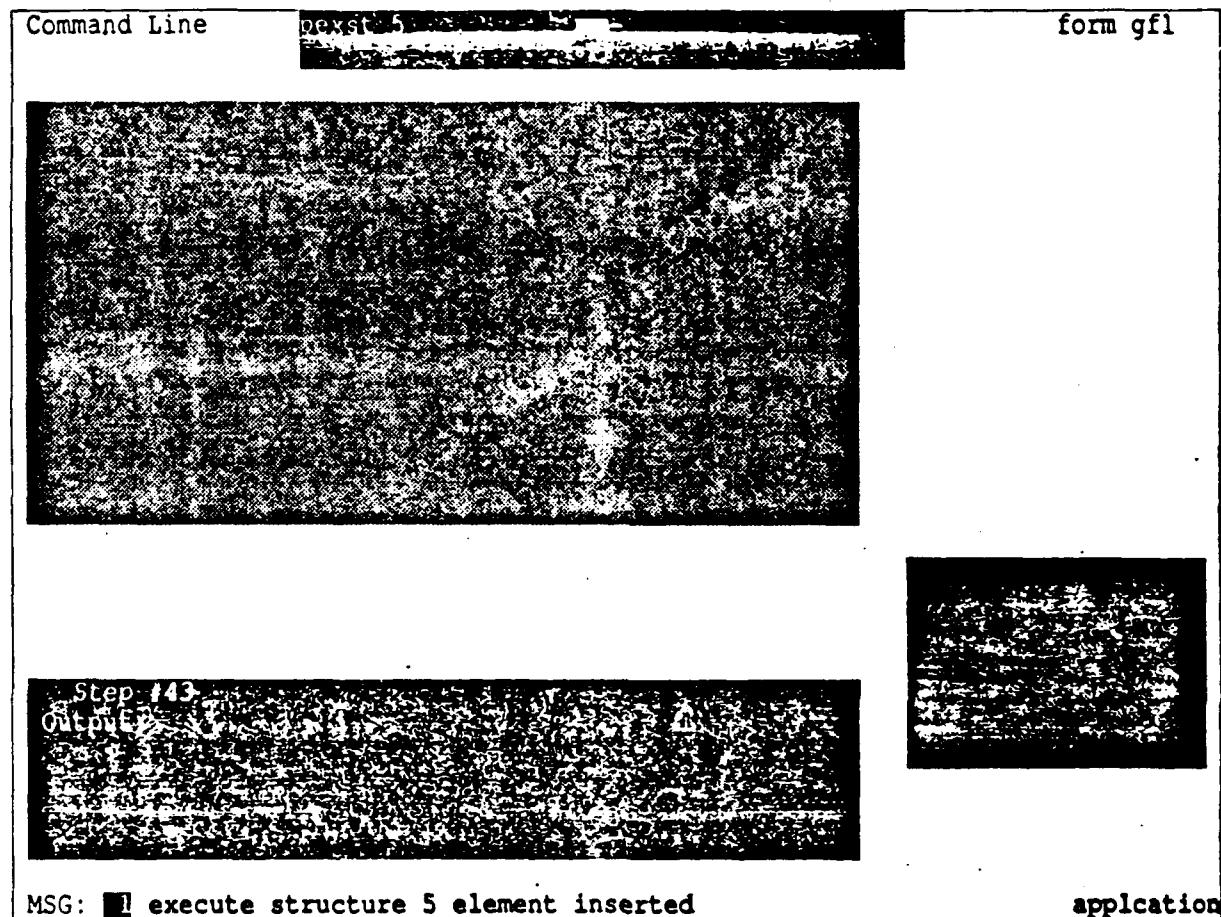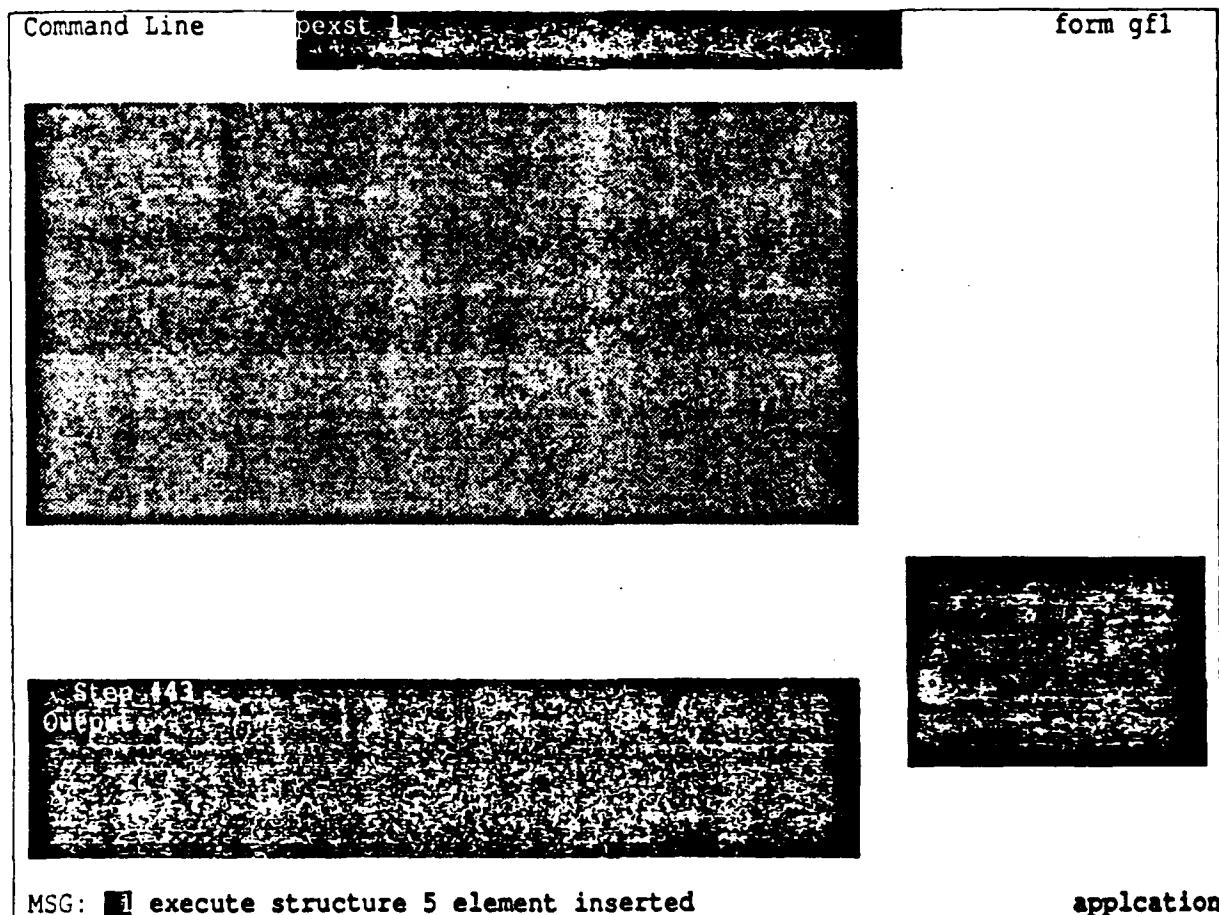
Figure 5-44b   (AFTER)

Figure 5-45a   (BEFORE)

Figure 5-45b   (AFTER)

Figure 5-46a   (BEFORE)
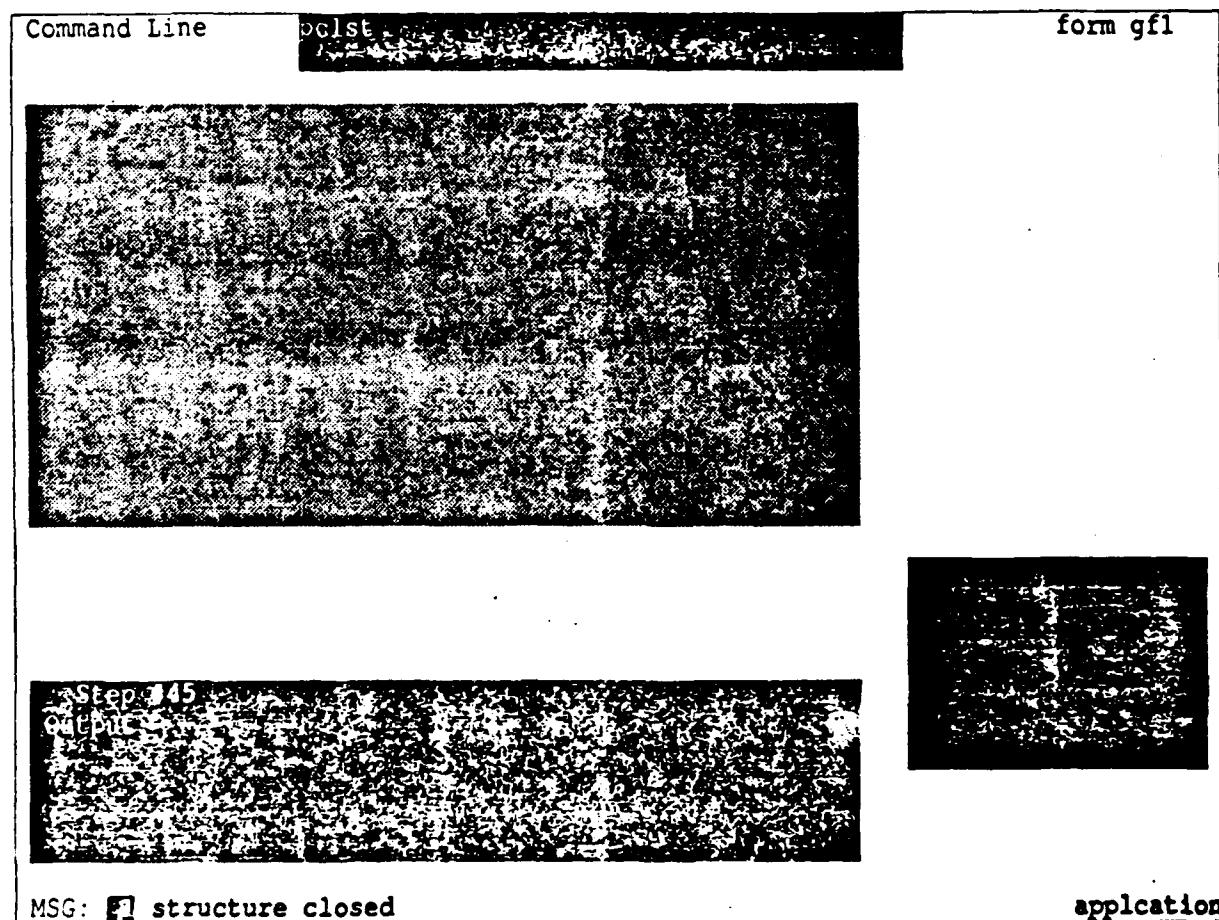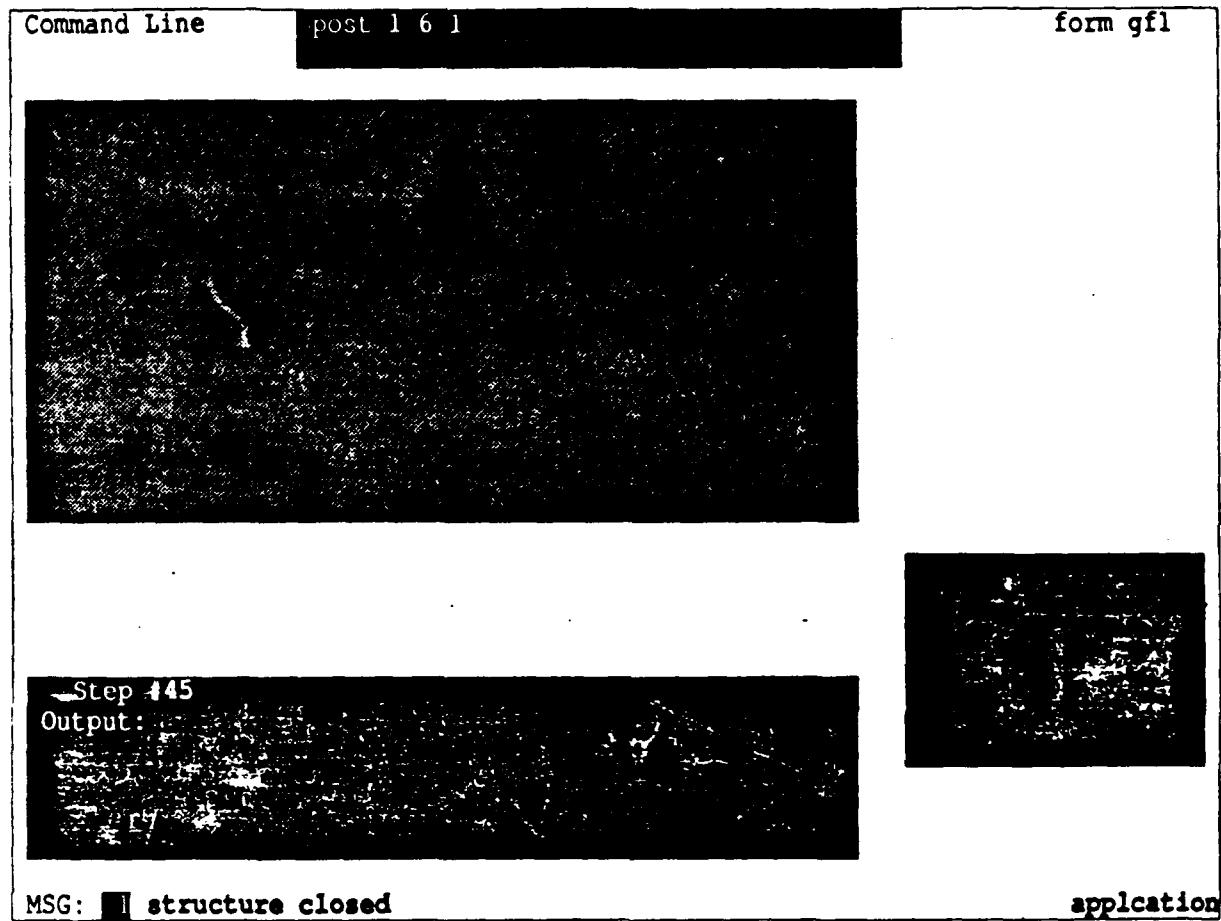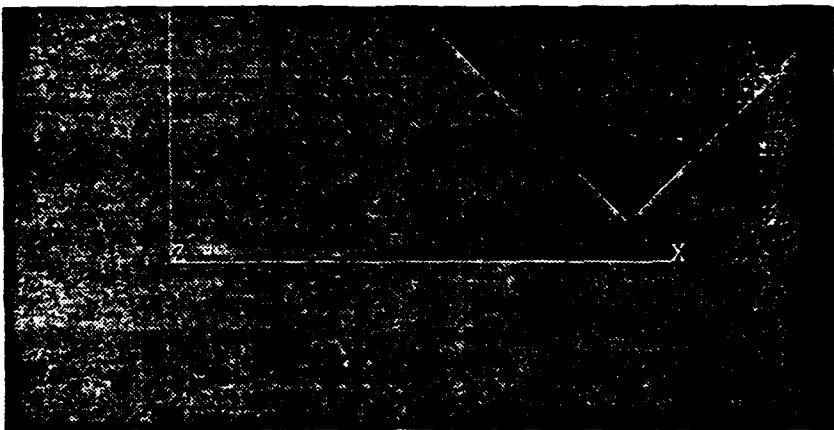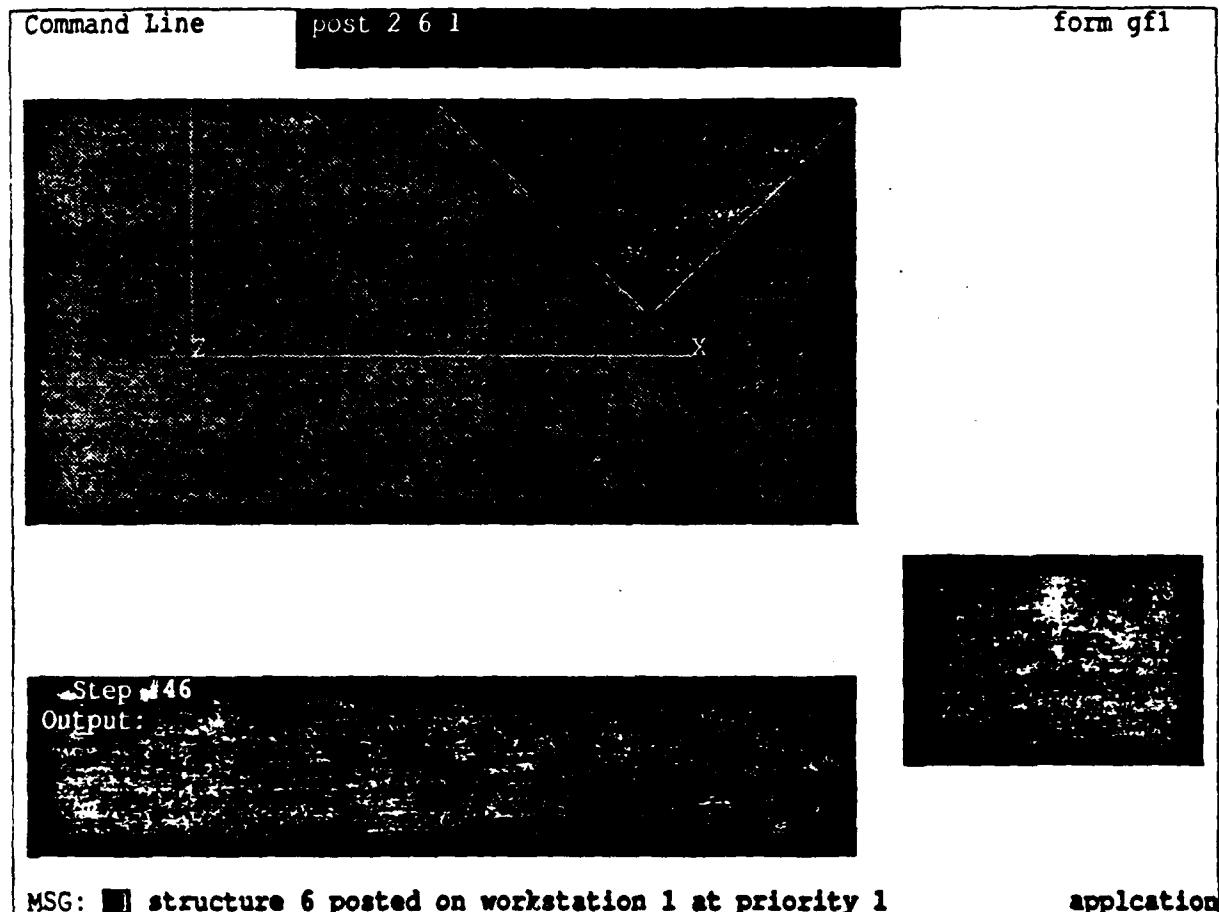
Figure 5-46b   (AFTER)

Figure 5-47a   (BEFORE)
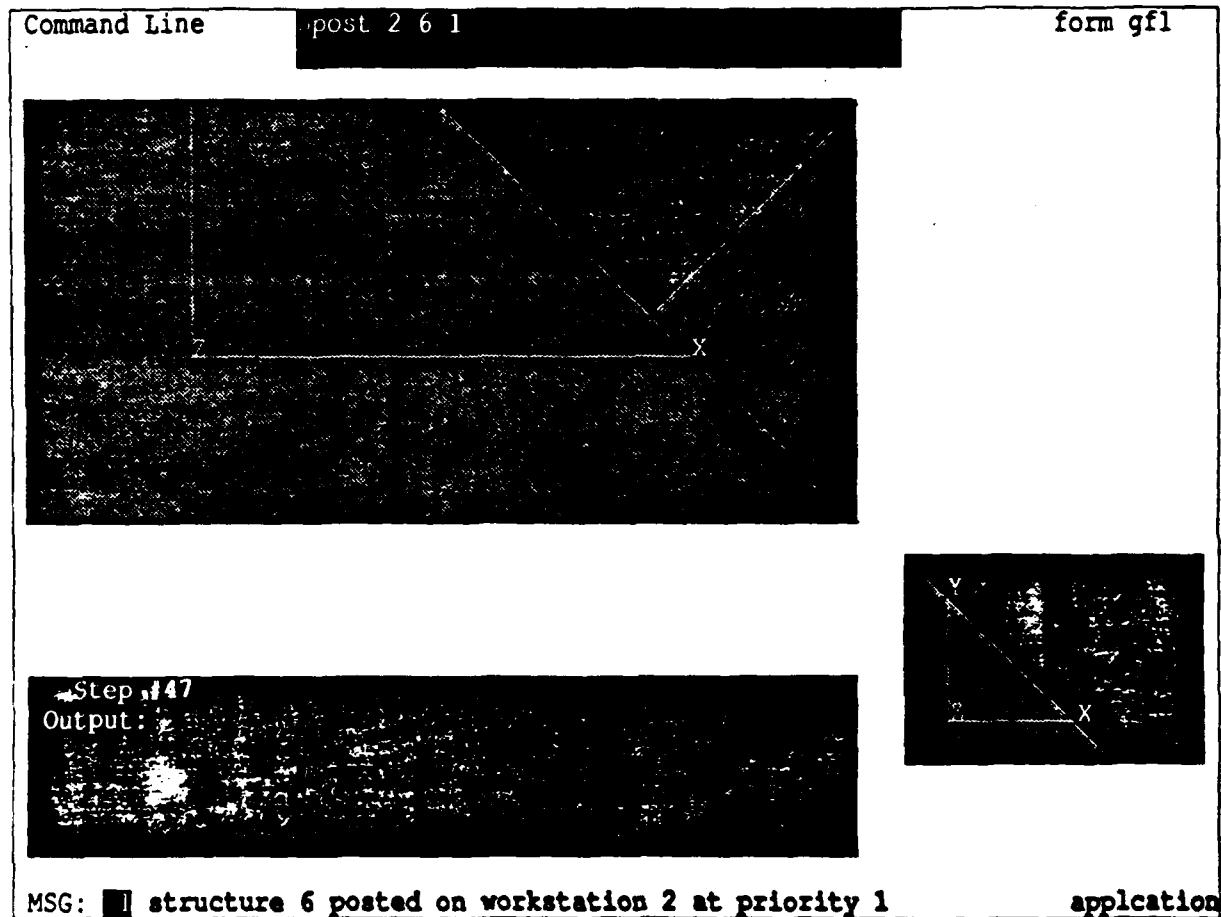
Figure 5-47b   (AFTER)

Figure 5-48a   (BEFORE)
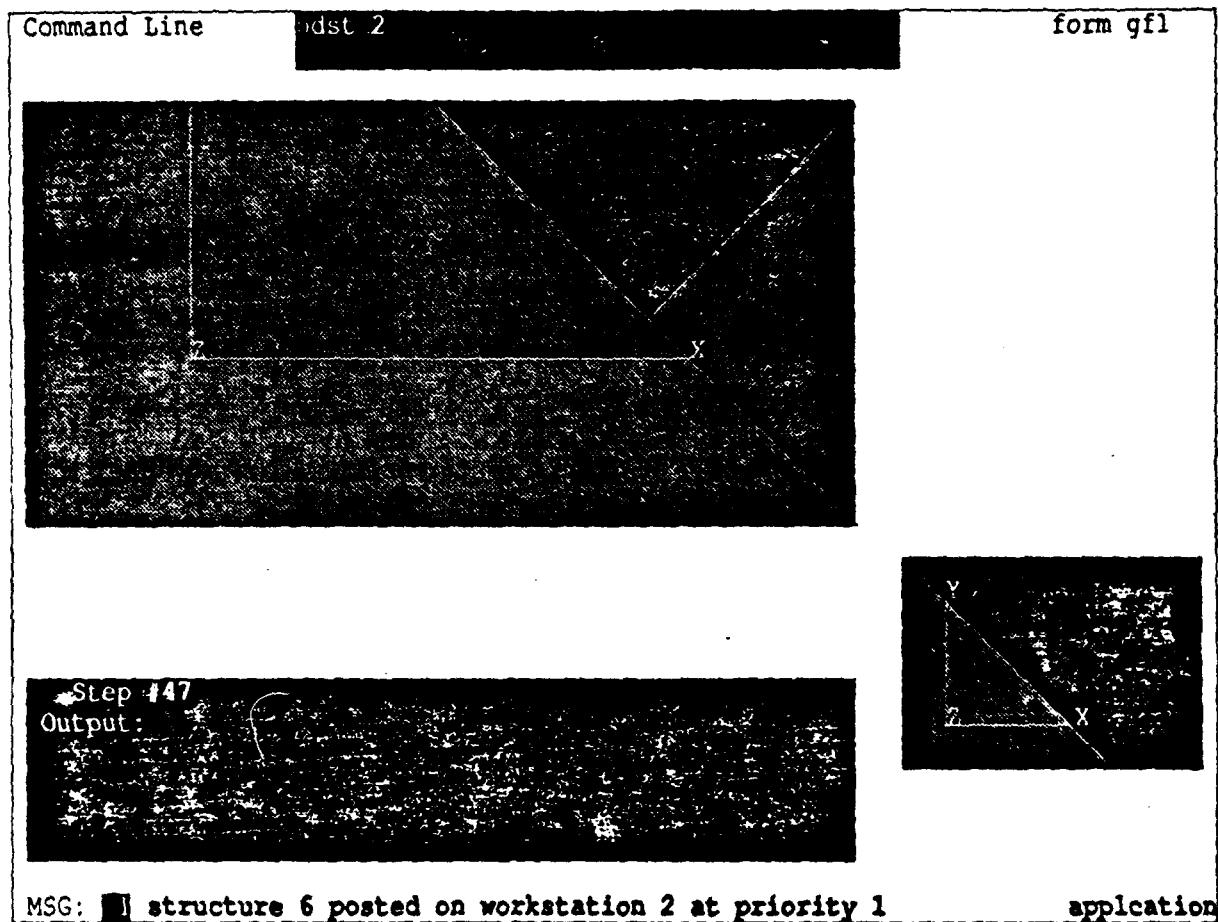
Figure 5-48b (AFTER)
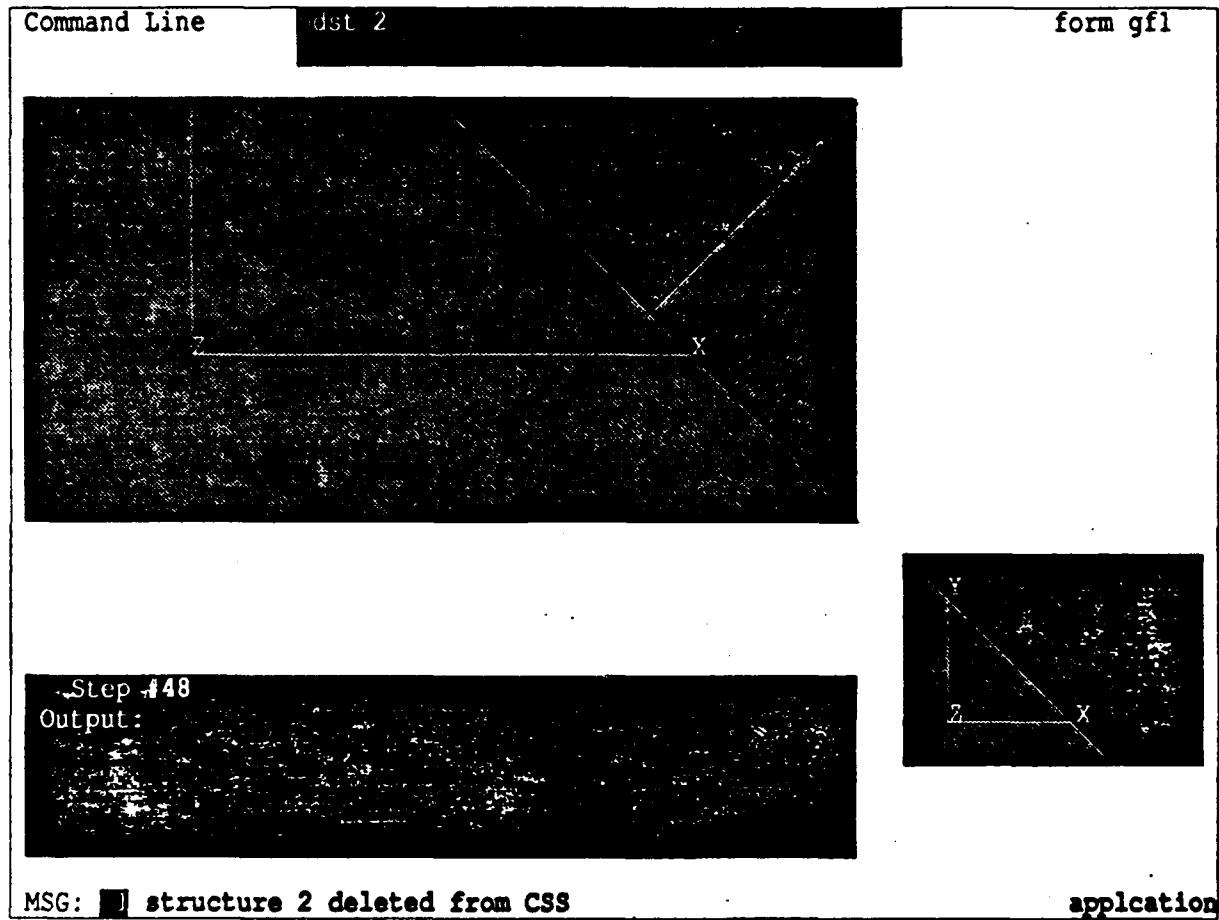
Figure 5-49a   (BEFORE)

Figure 5-49b   (AFTER)
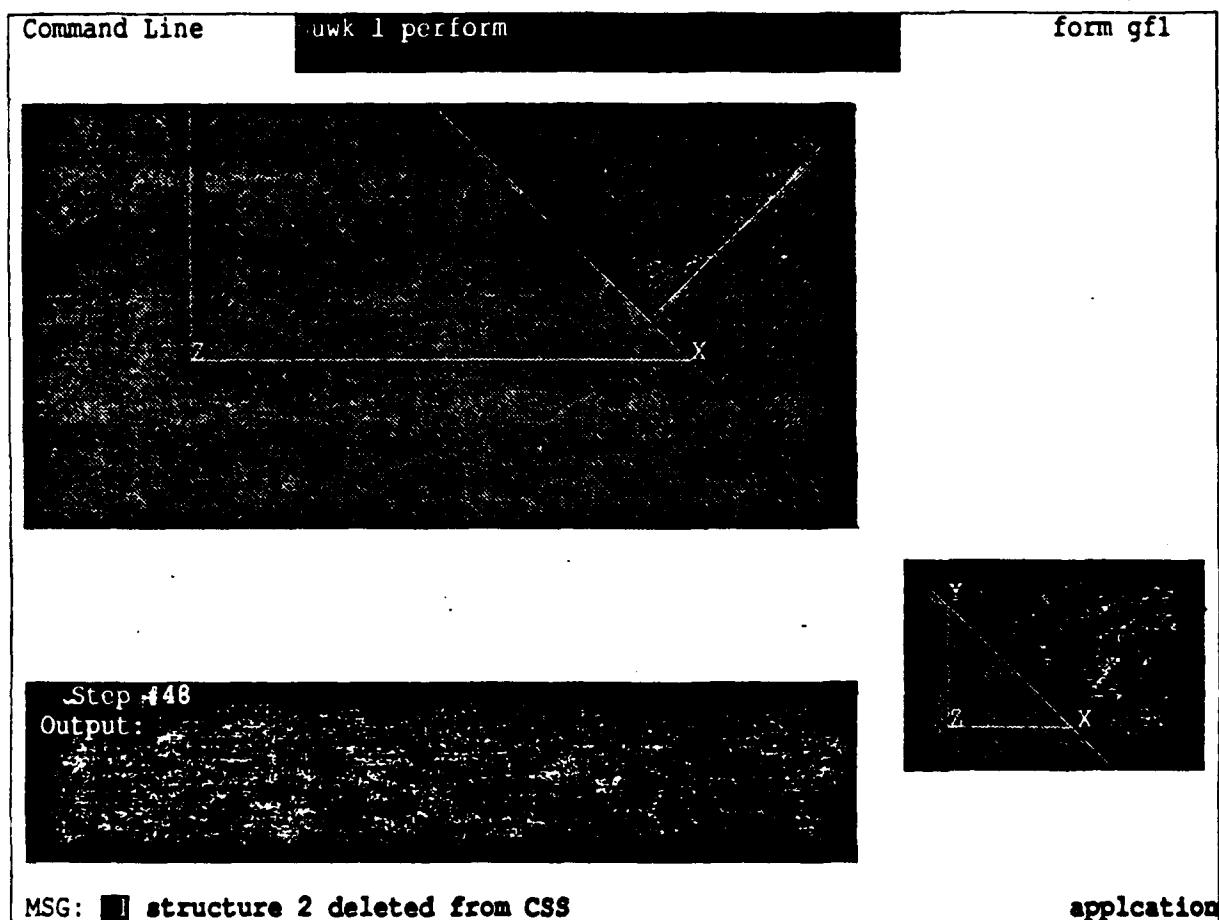
Figure 5-50a   (BEFORE)
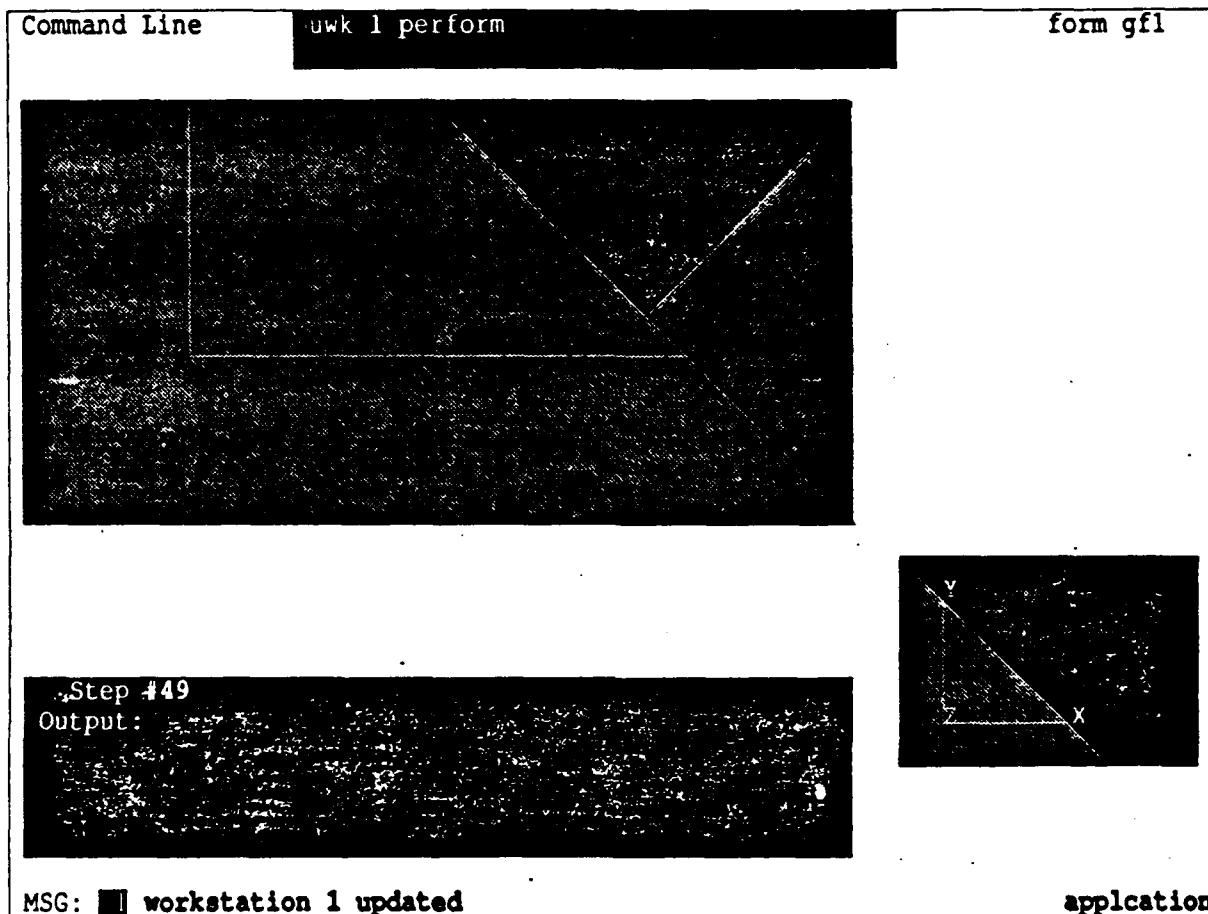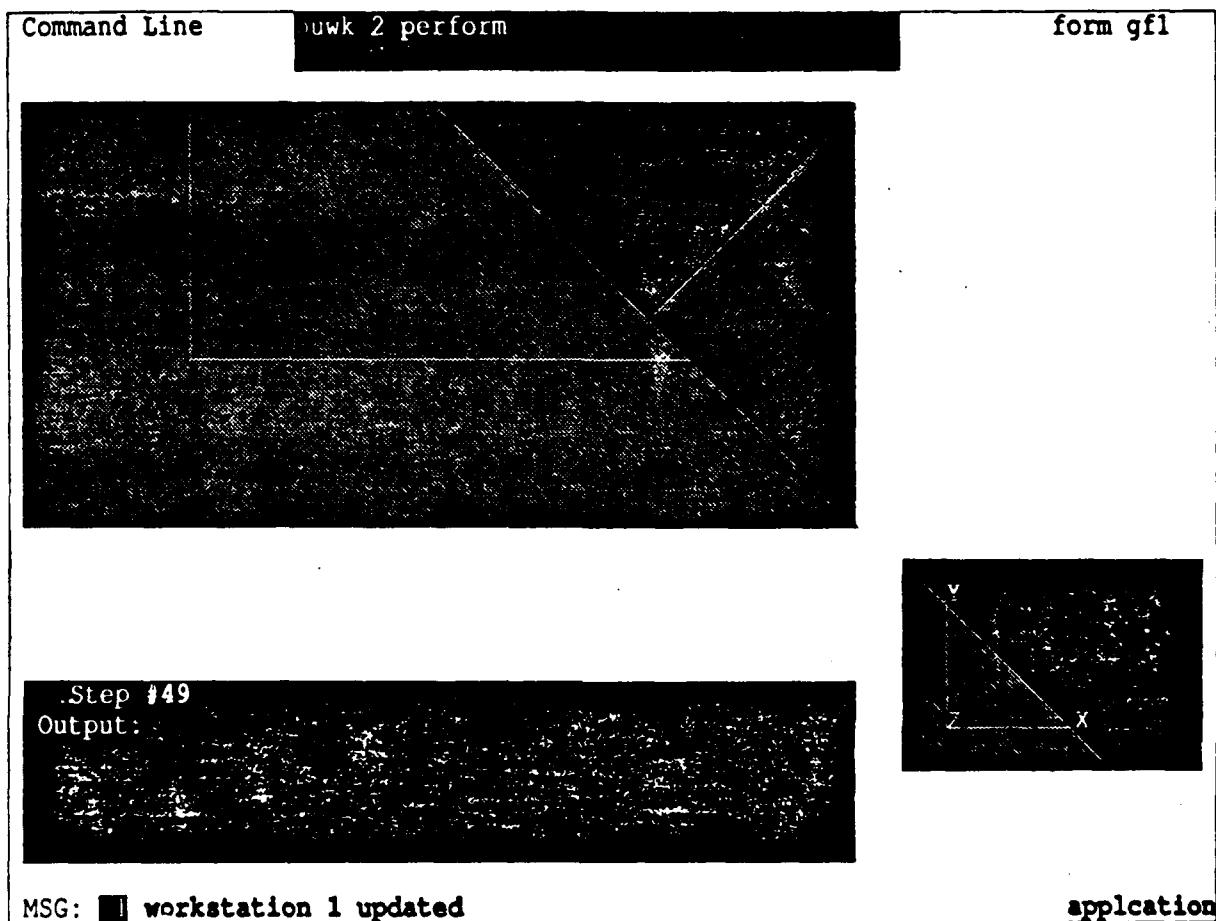
Figure 5-50b   (AFTER)

Figure 5-51a   (BEFORE)
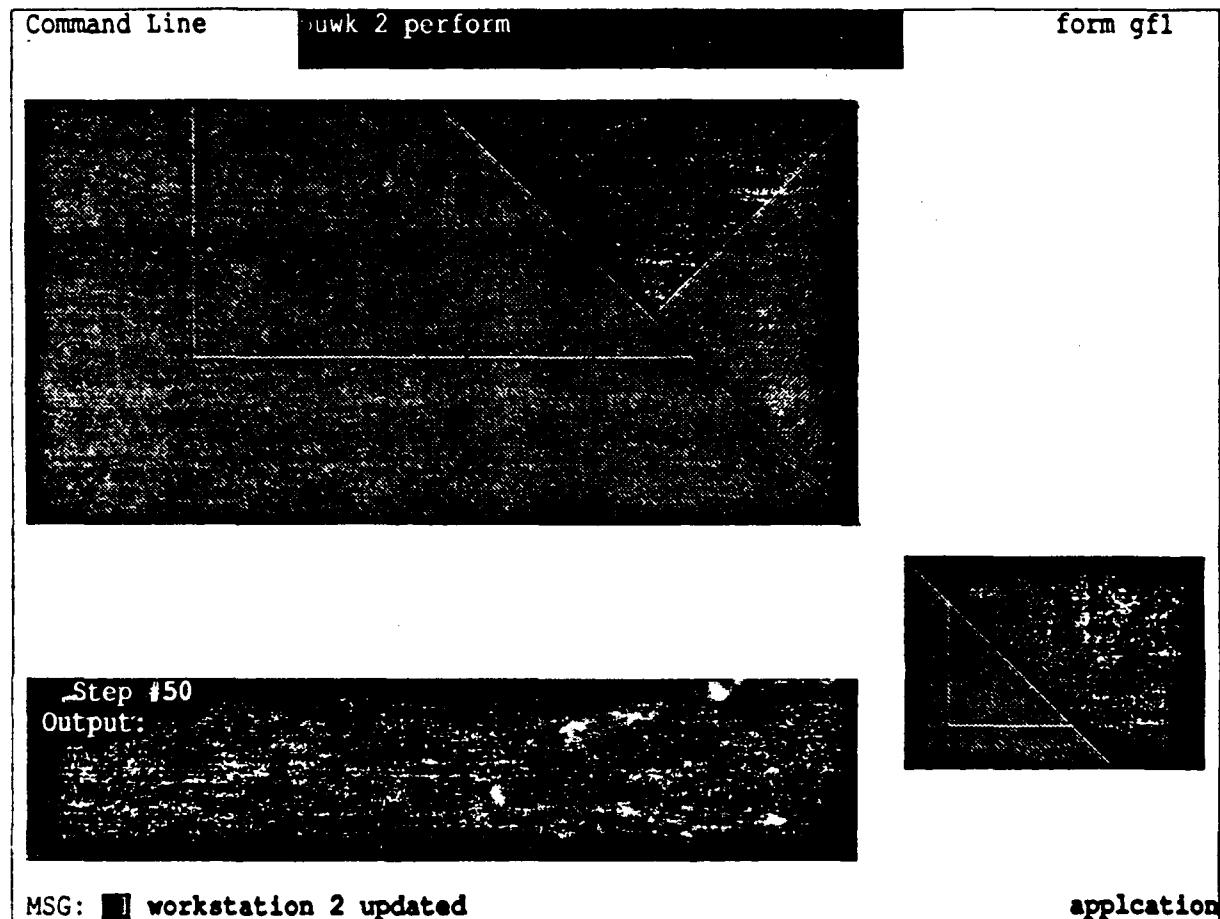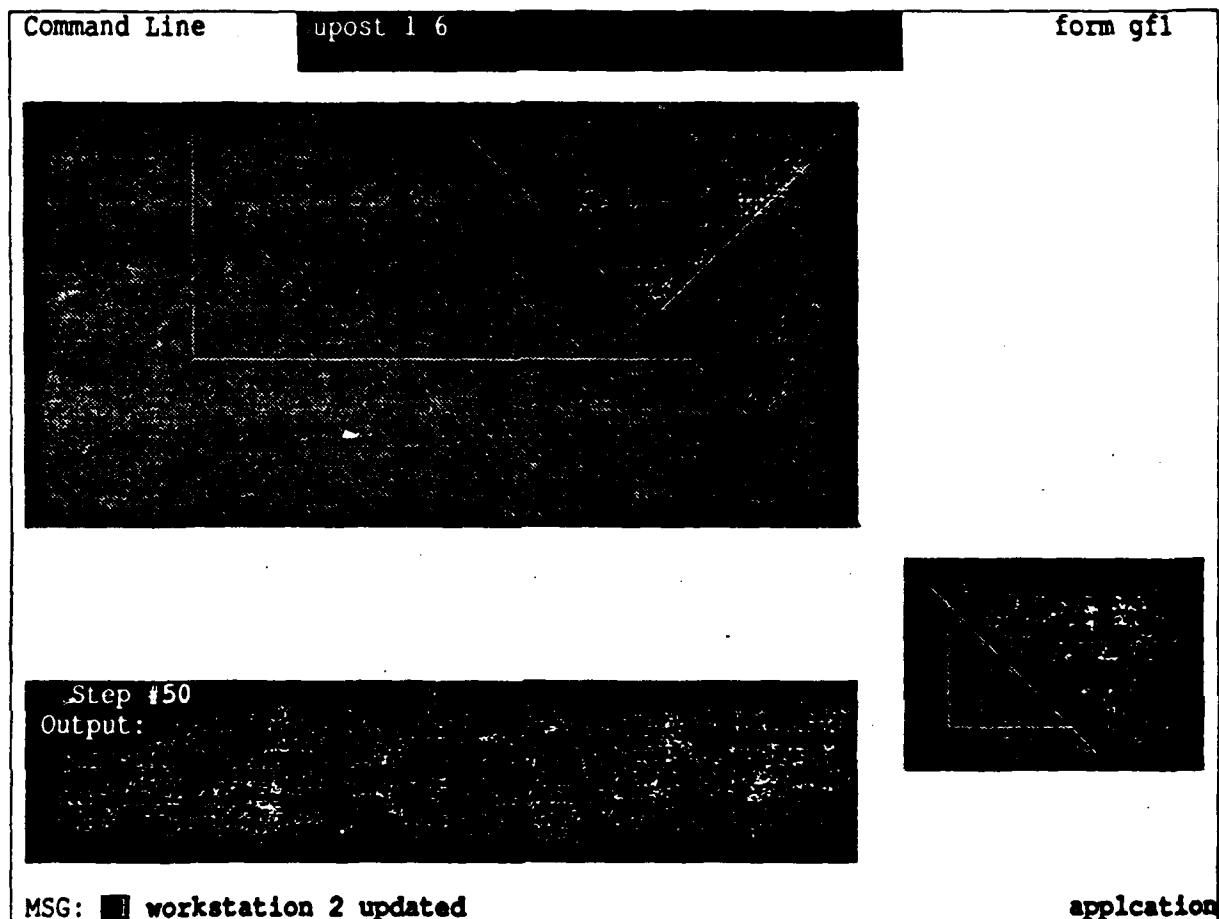
Figure 5-51b   (AFTER)

Figure 5-52a   (BEFORE)
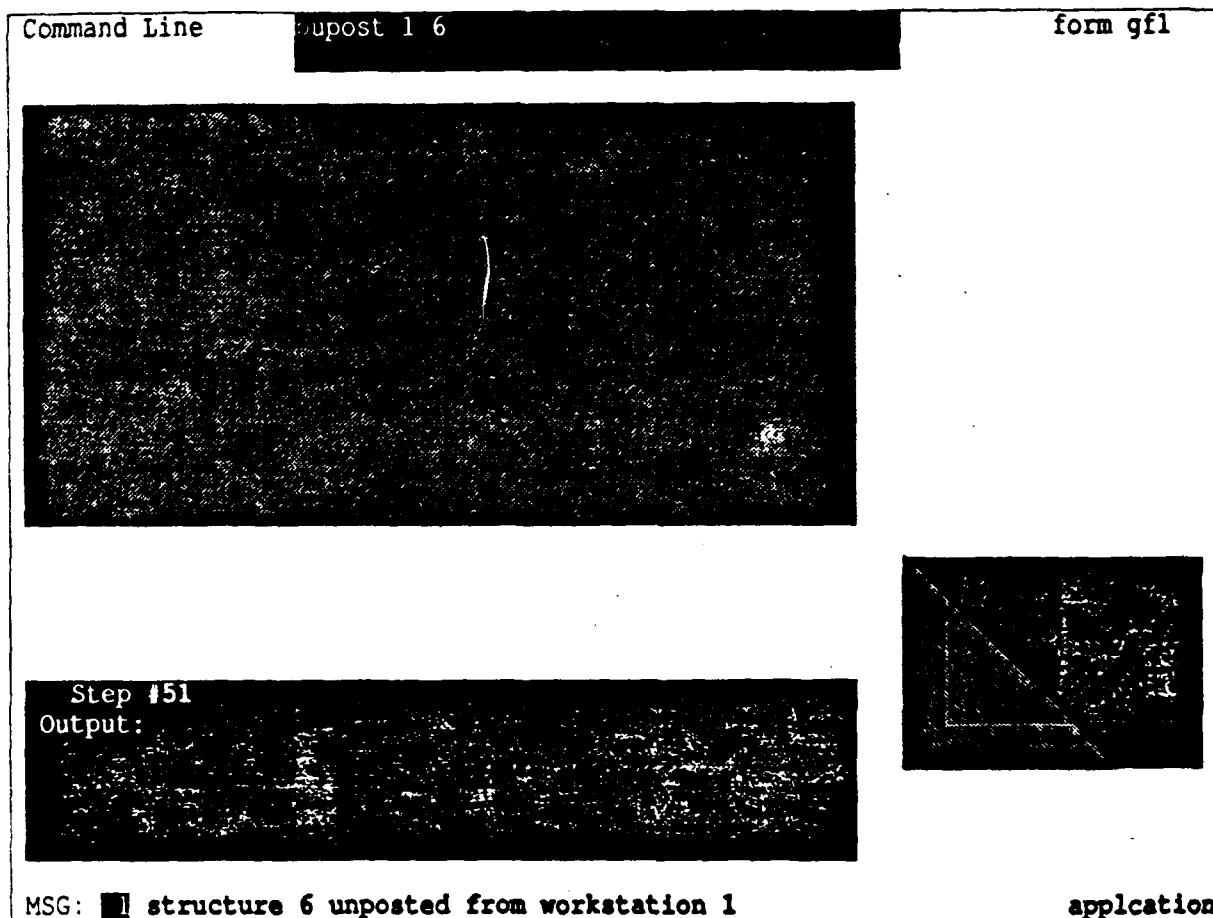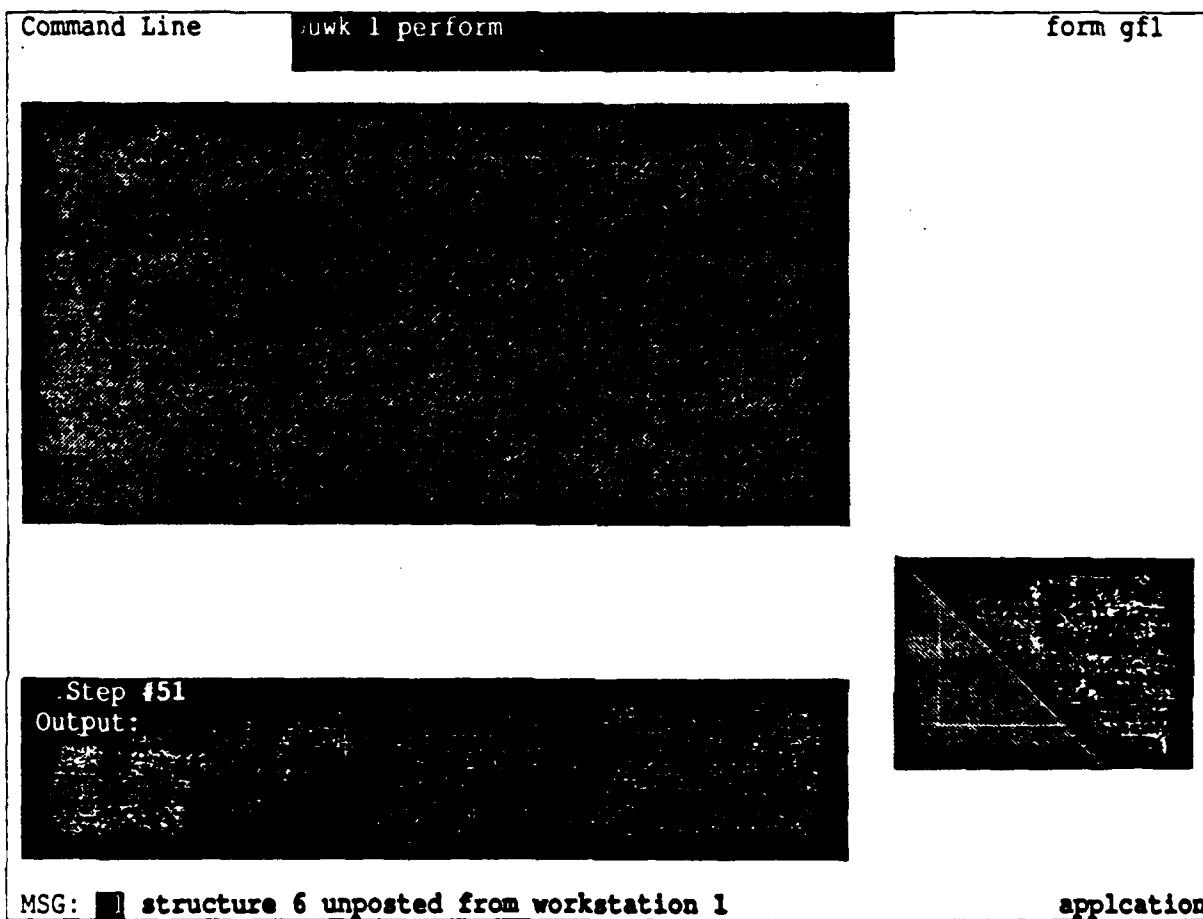
Figure 5-52b   (AFTER)

Figure 5-53a   (BEFORE)
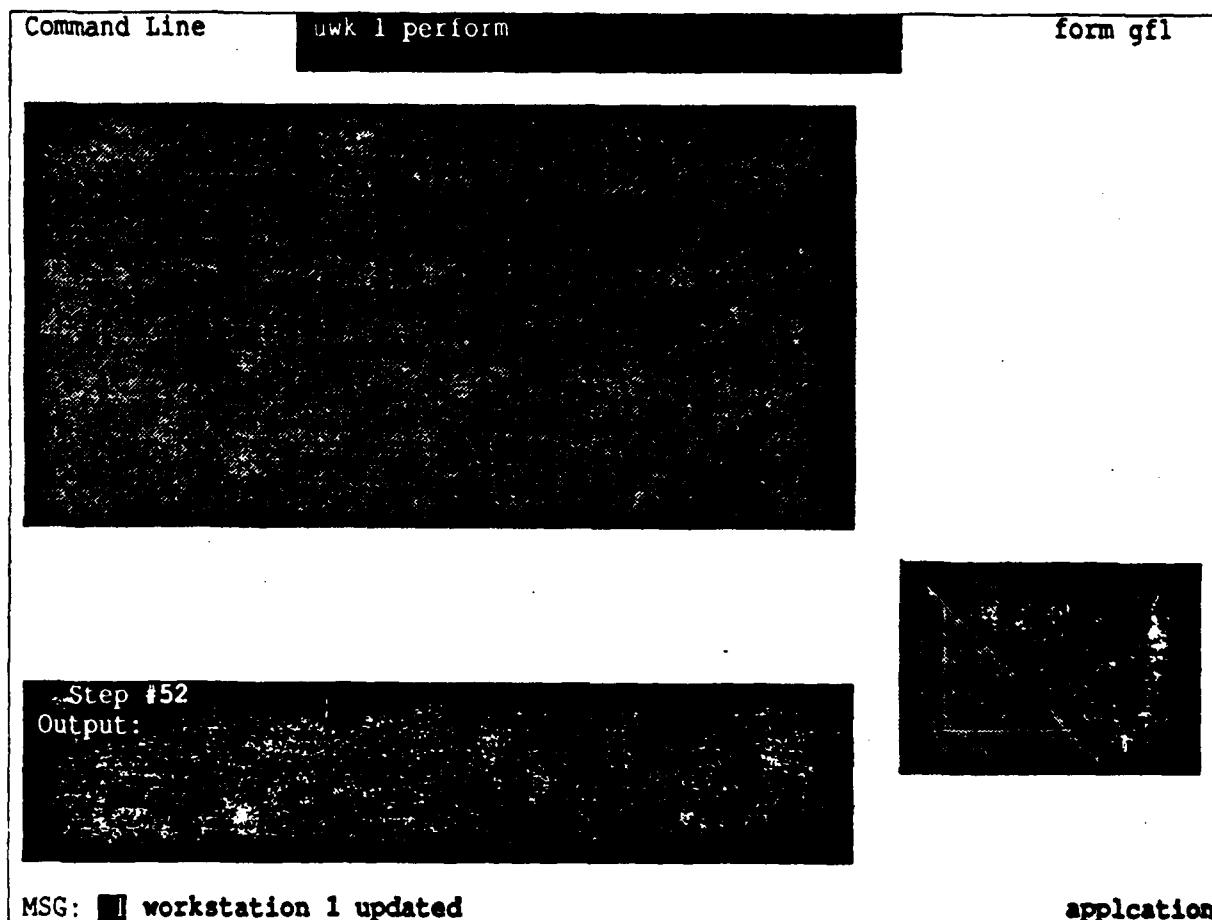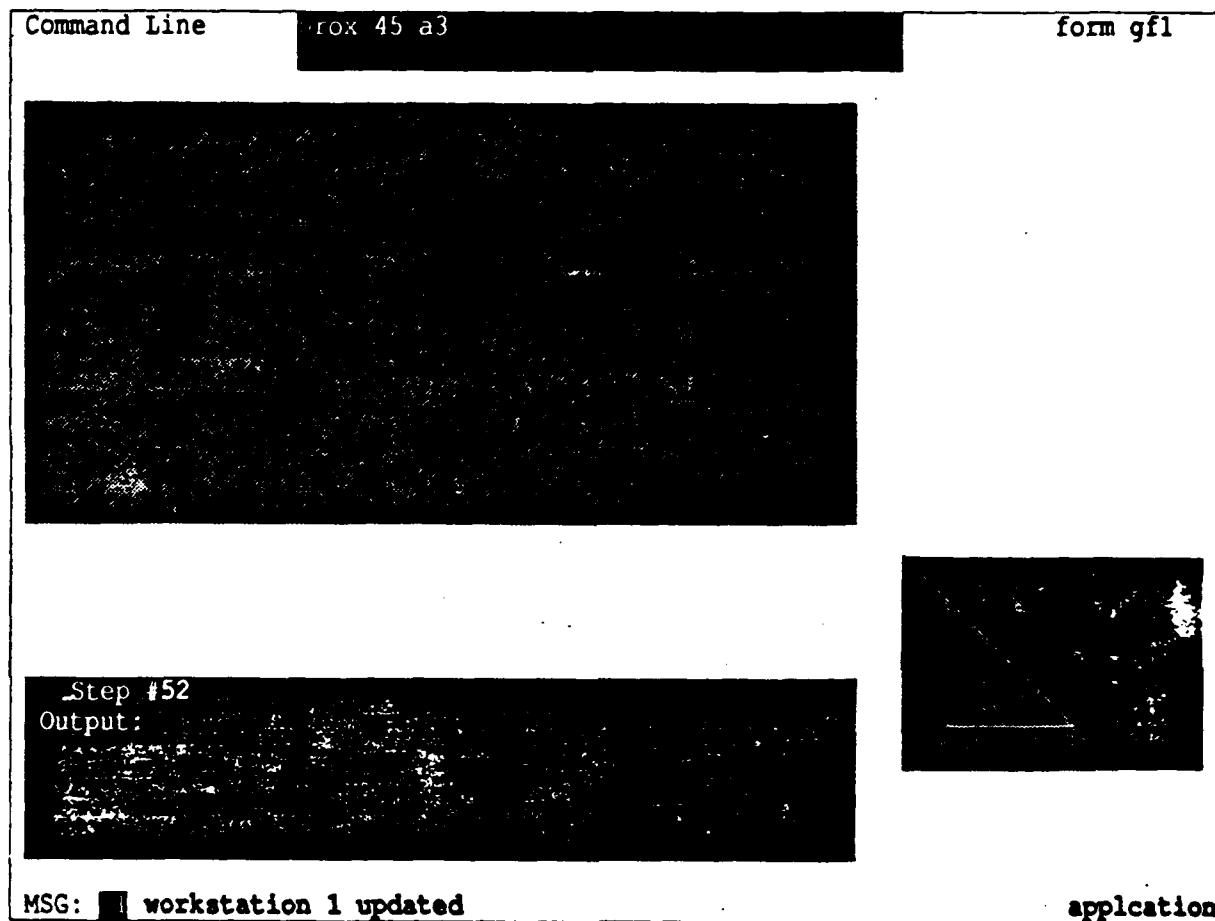
Figure 5-53b   (AFTER)

Figure 5-54a   (BEFORE)

Command Line   rox 45 a3   form gfl

_Step #53
Output:

MSG: ▮ 3-D rotation matrix created   applcation
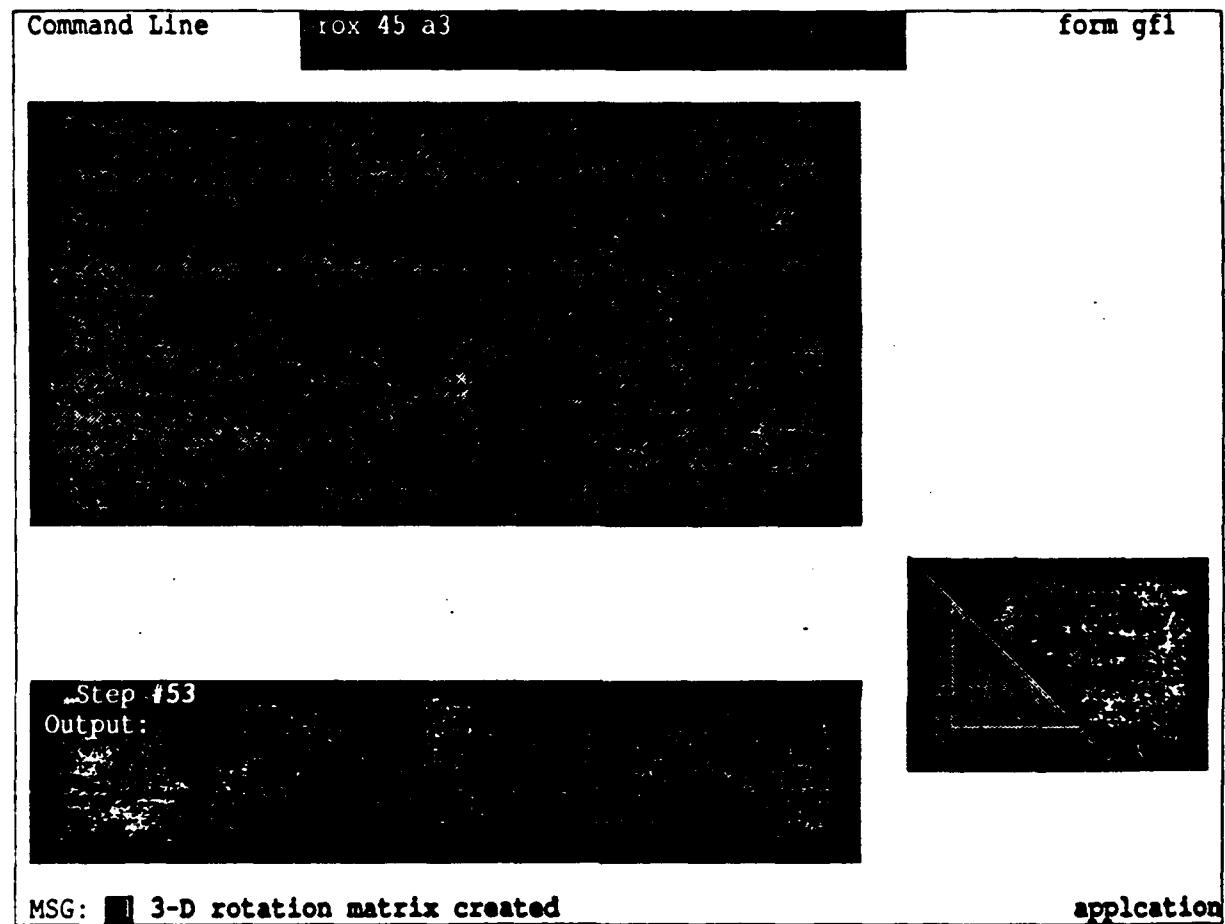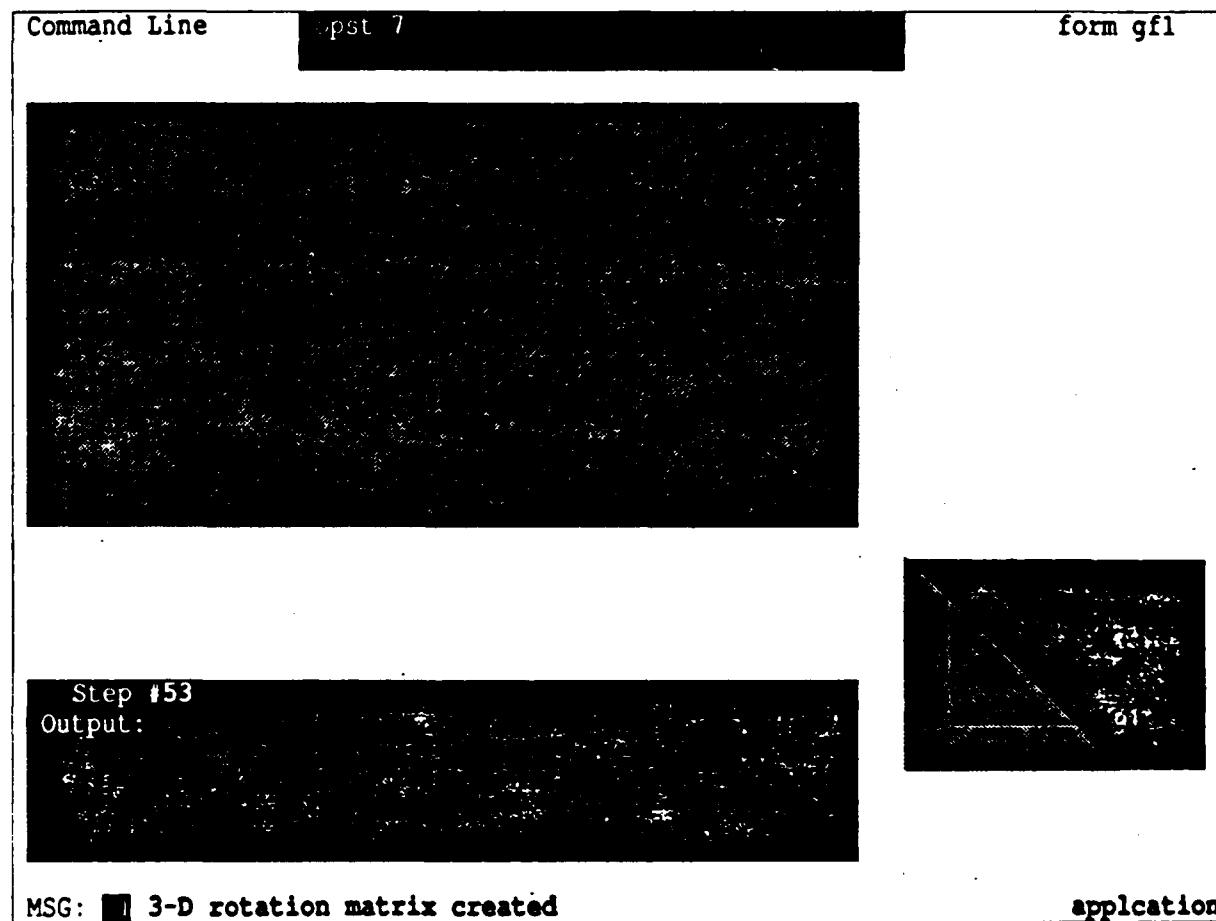
Figure 5-54b   (AFTER)

Figure 5-55a   (BEFORE)
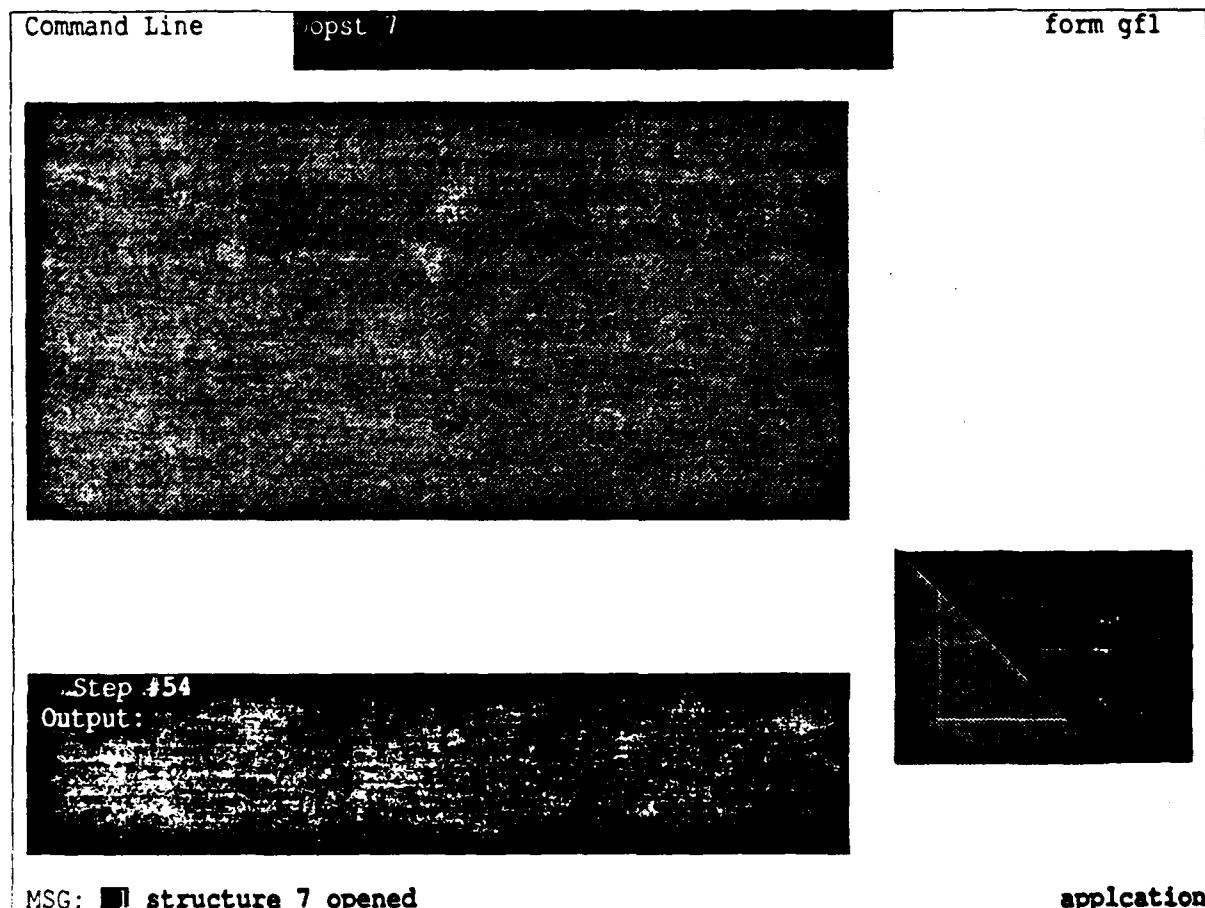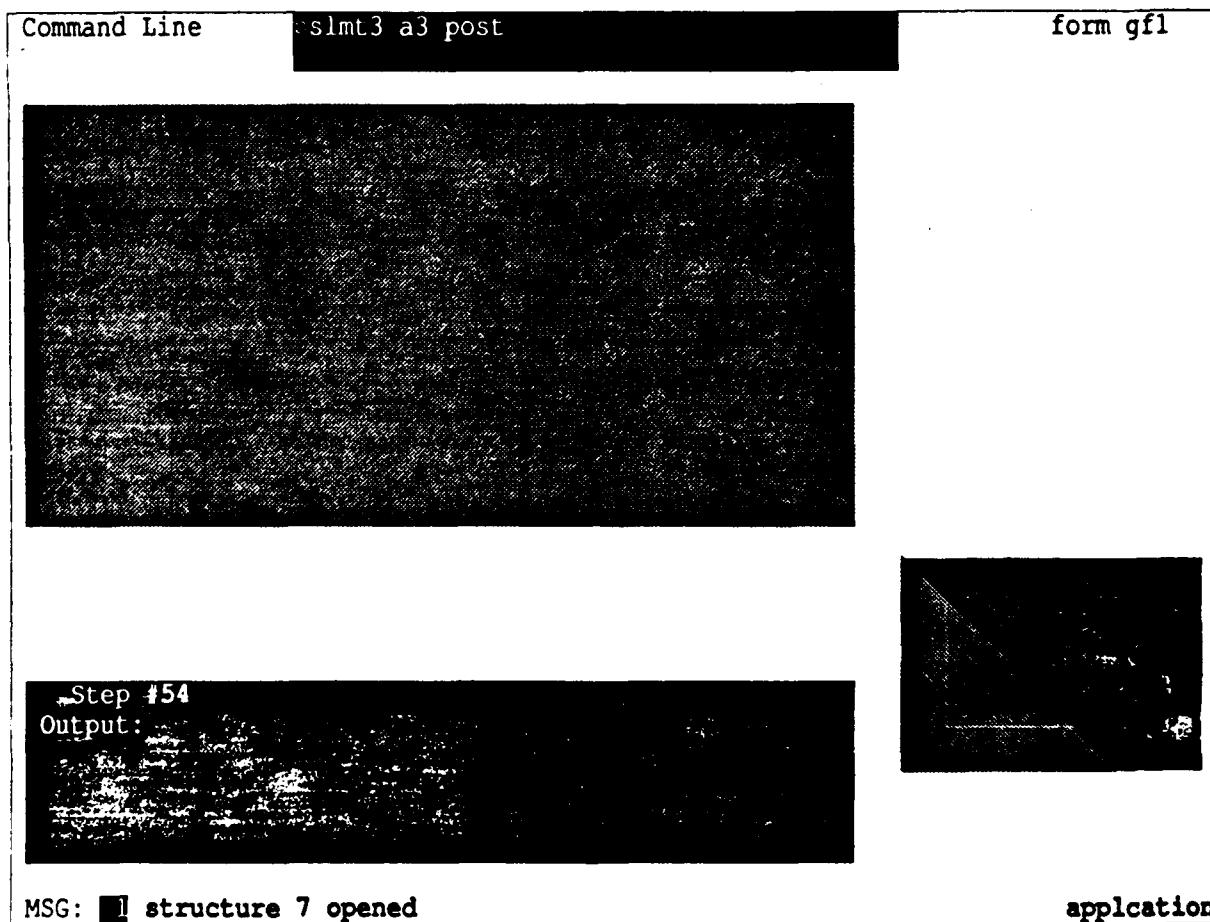
Figure 5-55b (AFTER)

Figure 5-56a   (BEFORE)

```
Command Line        slmt3 a3 post                          form gfl


                                          Step #55
                                          Output:

MSG: ▋ 3-D transformation matrix element inserted            applcation
```
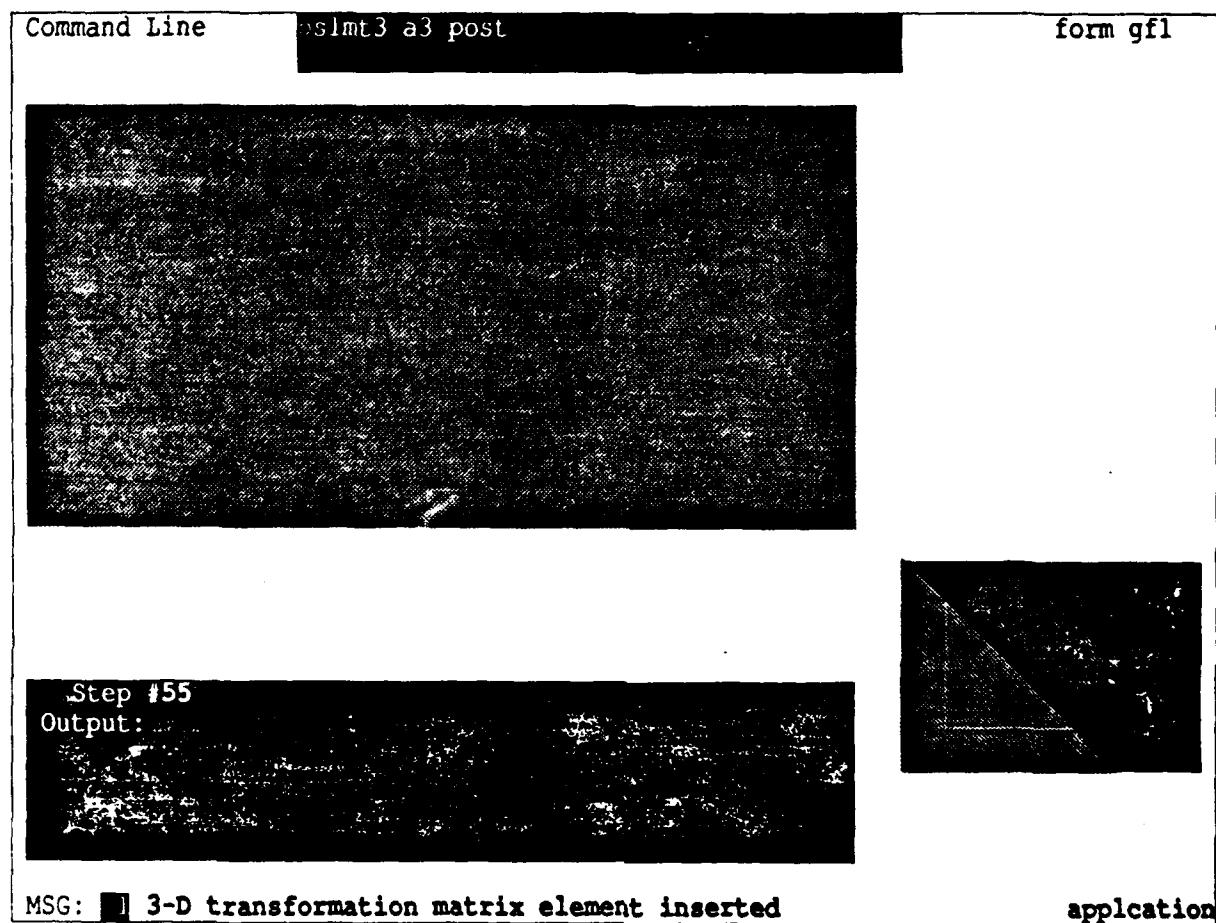
Figure 5-56b   (AFTER)

Figure 5-57a   (BEFORE)

Command Line   bexst 6   form gfl

Step #56
Output:

MSG: █1 execute structure 6 element inserted   applcation
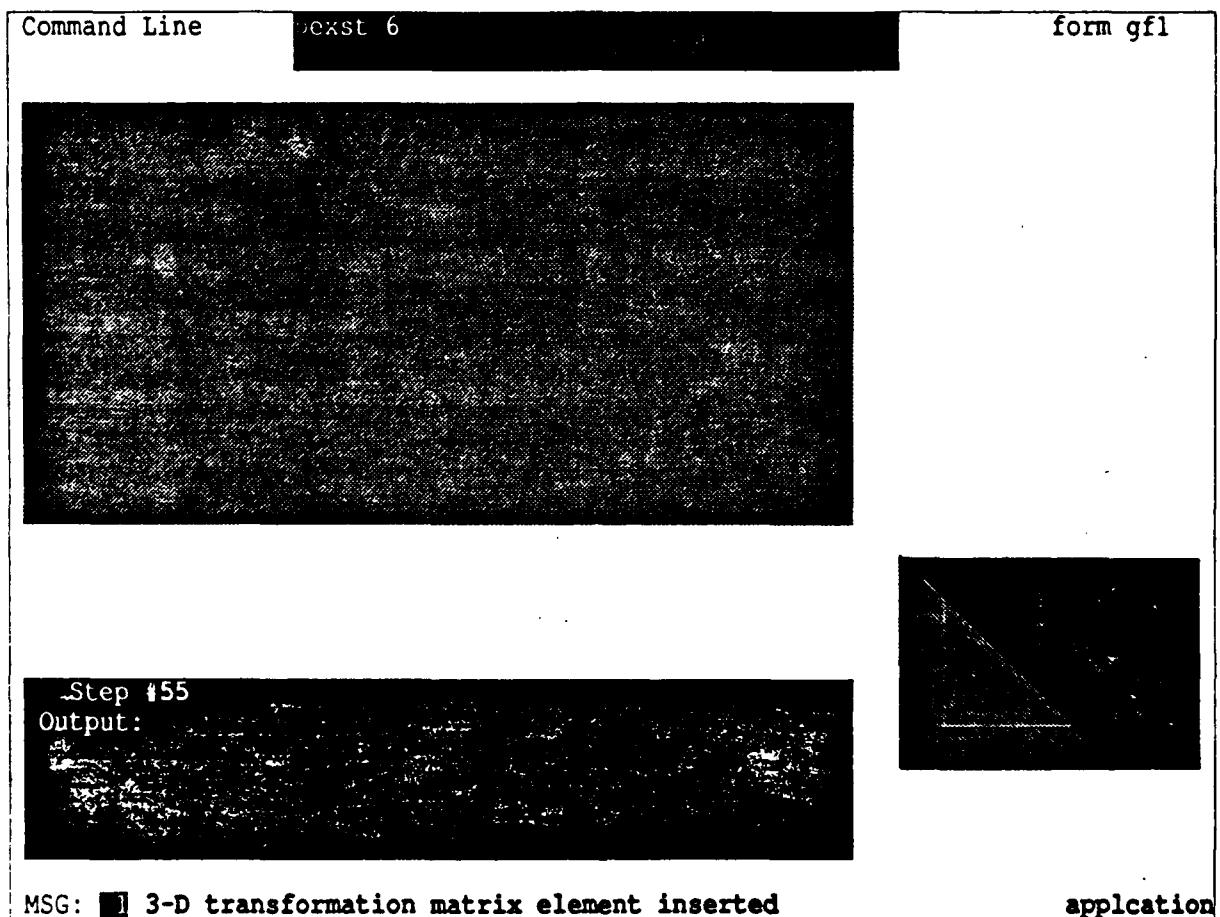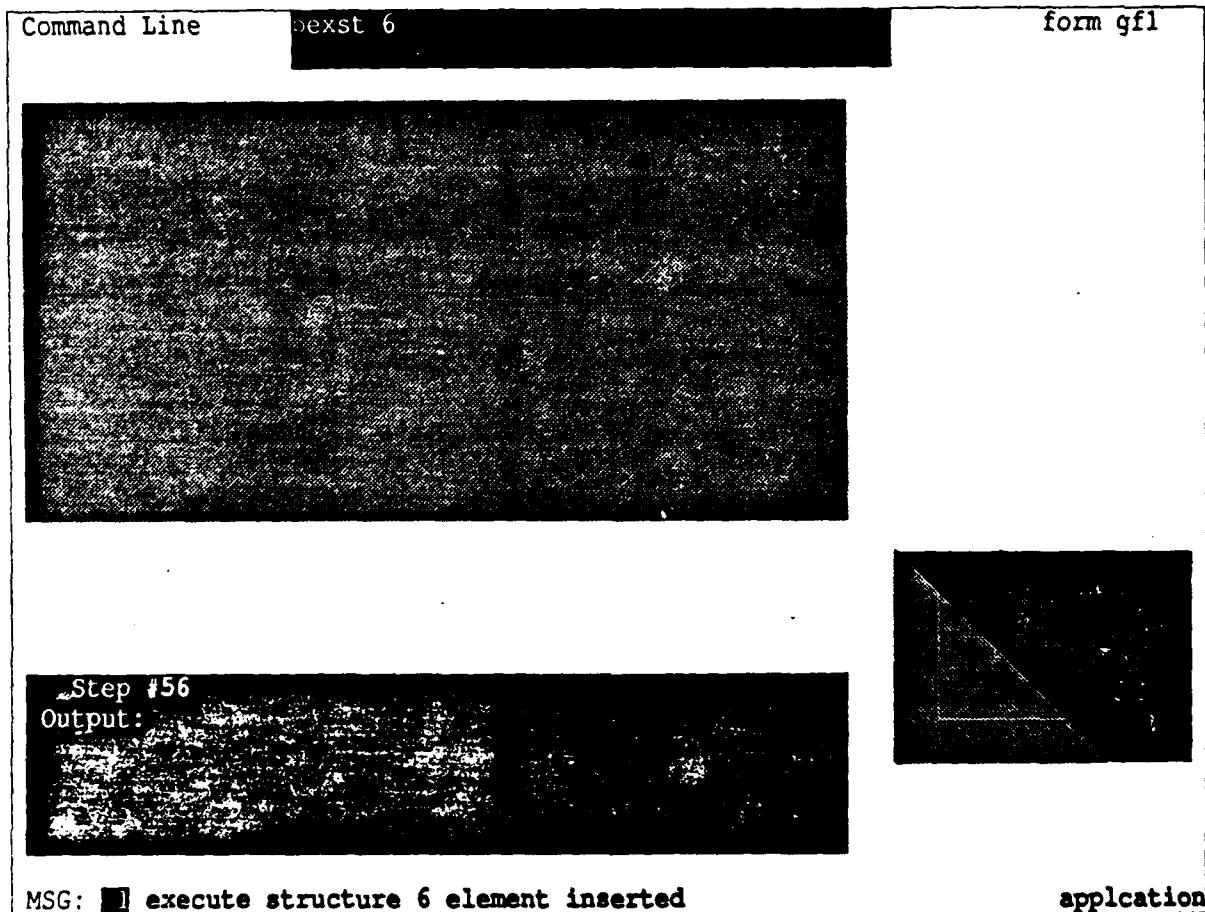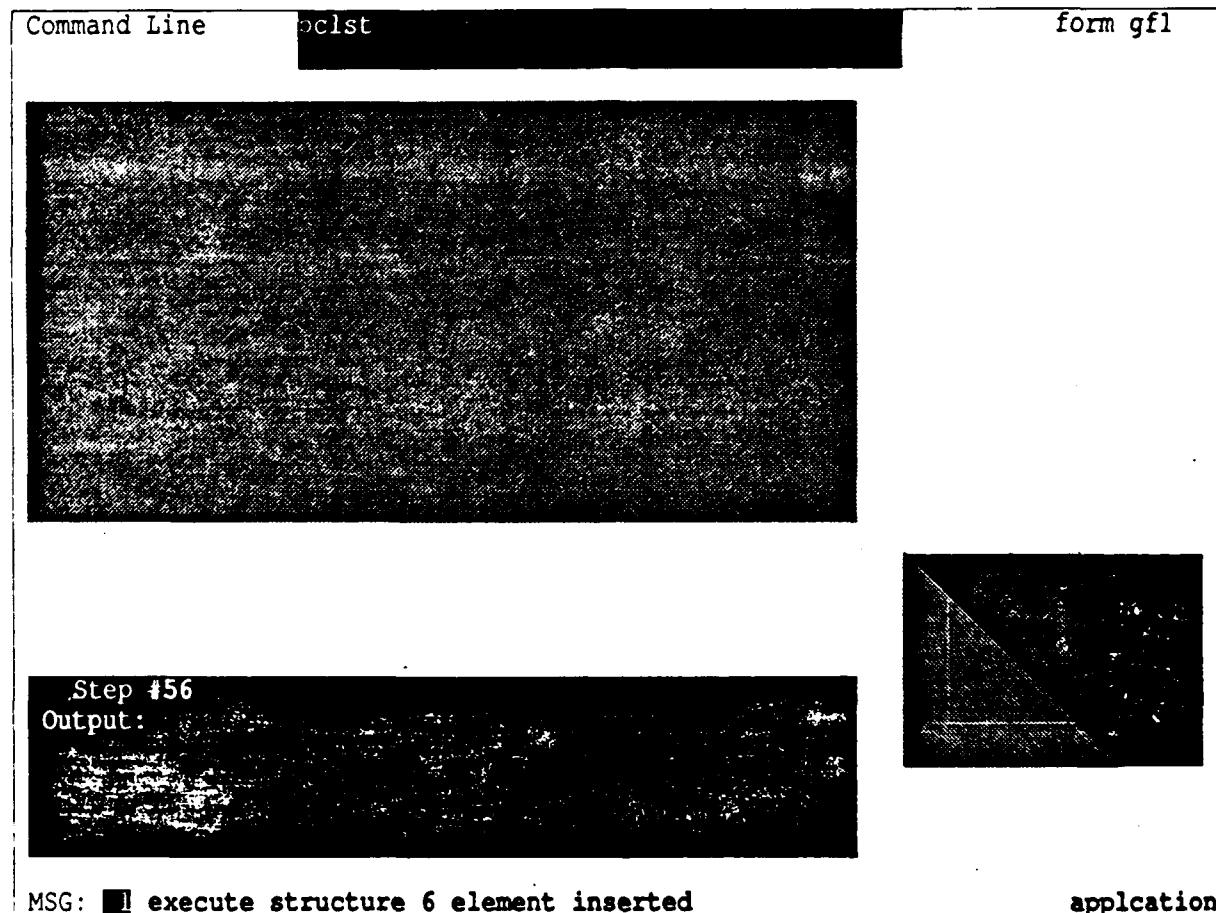
Figure 5-57b (AFTER)

Figure 5-58a (BEFORE)
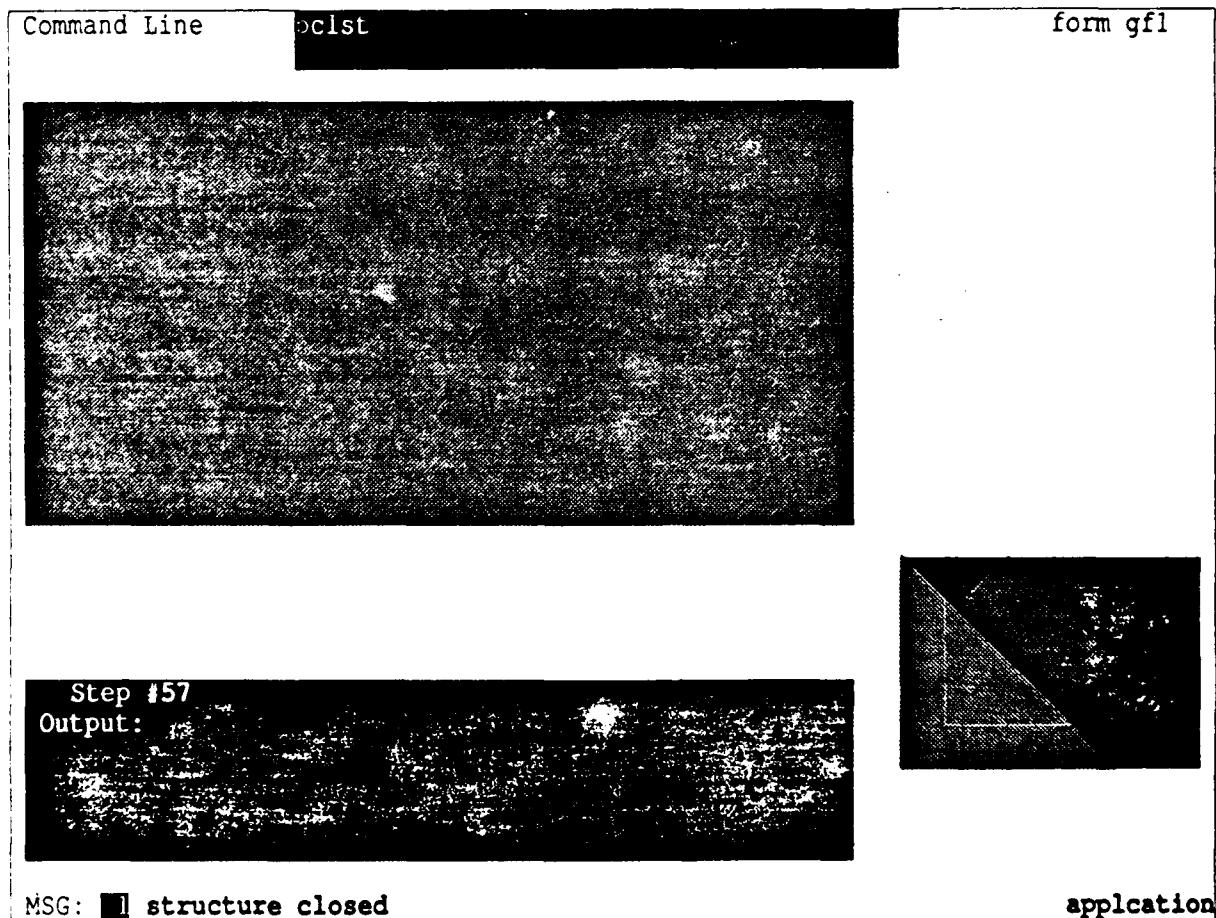
Figure 5-58b   (AFTER)
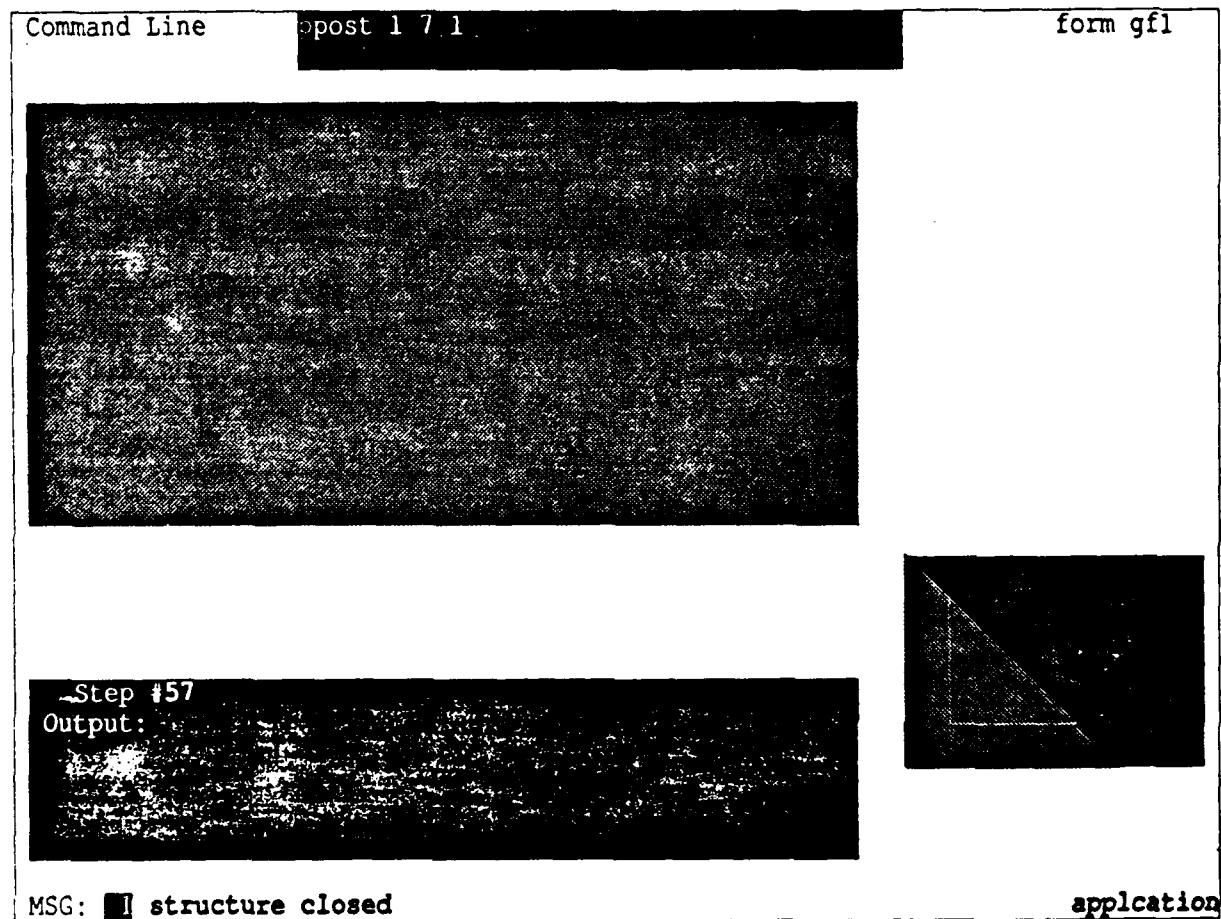
Figure 5-59a   (BEFORE)
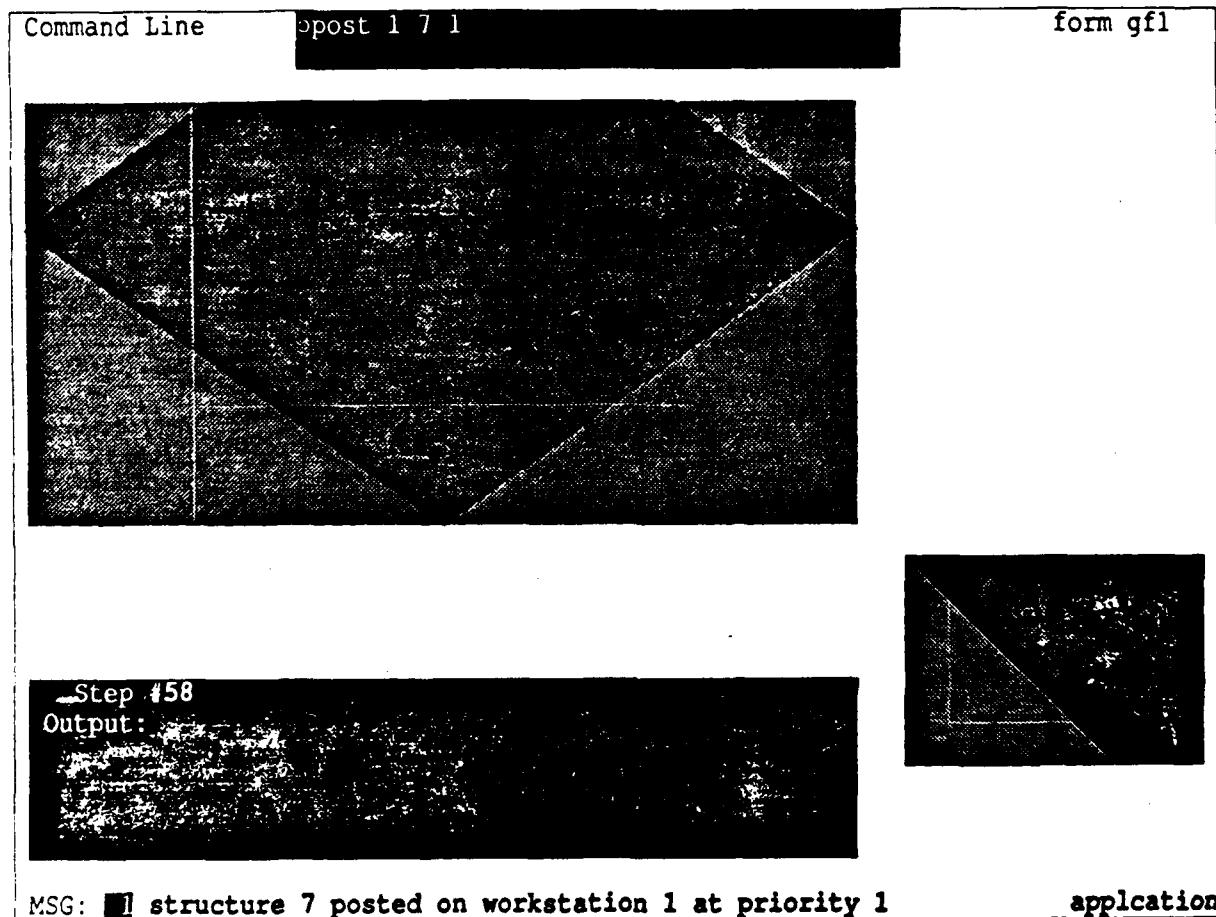
Figure 5-59b   (AFTER)
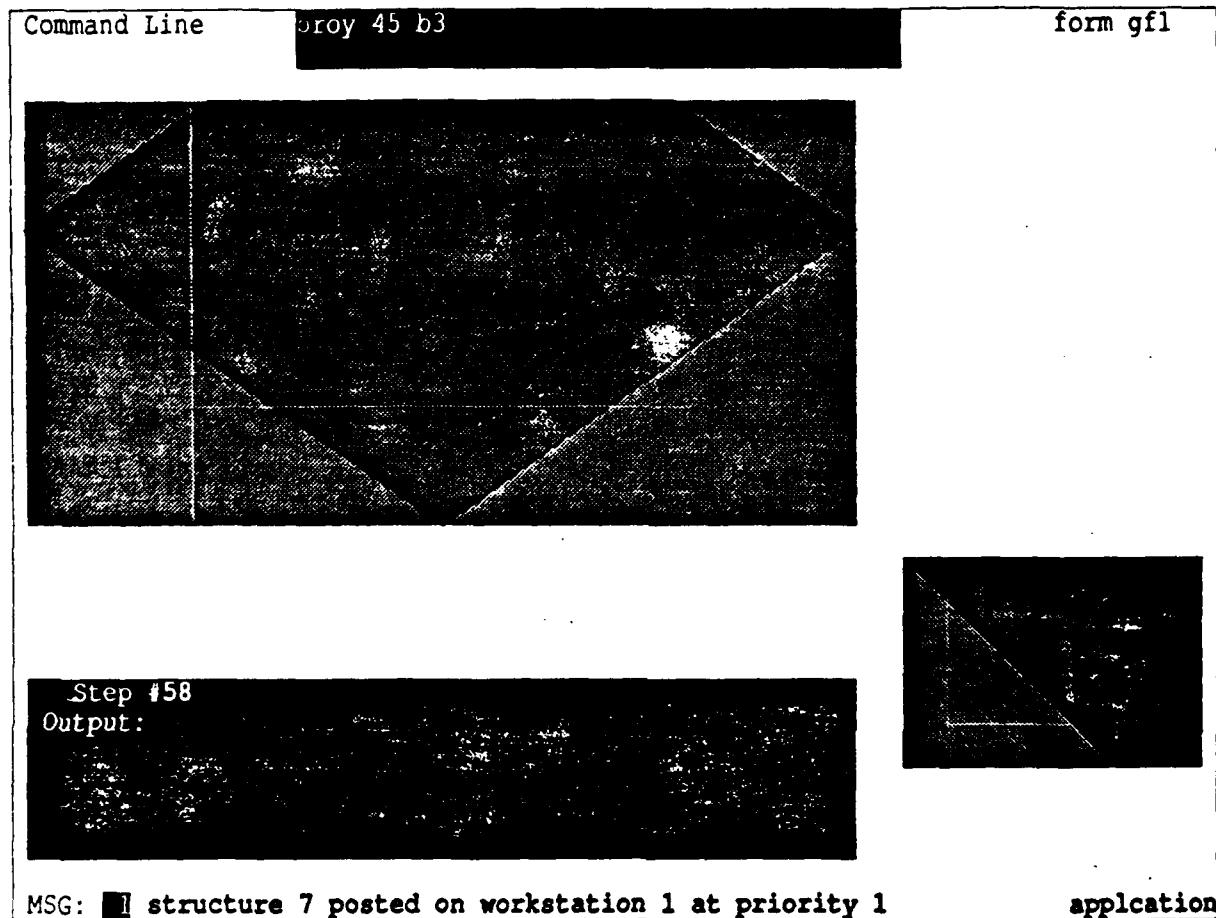
Figure 5-60a   (BEFORE)
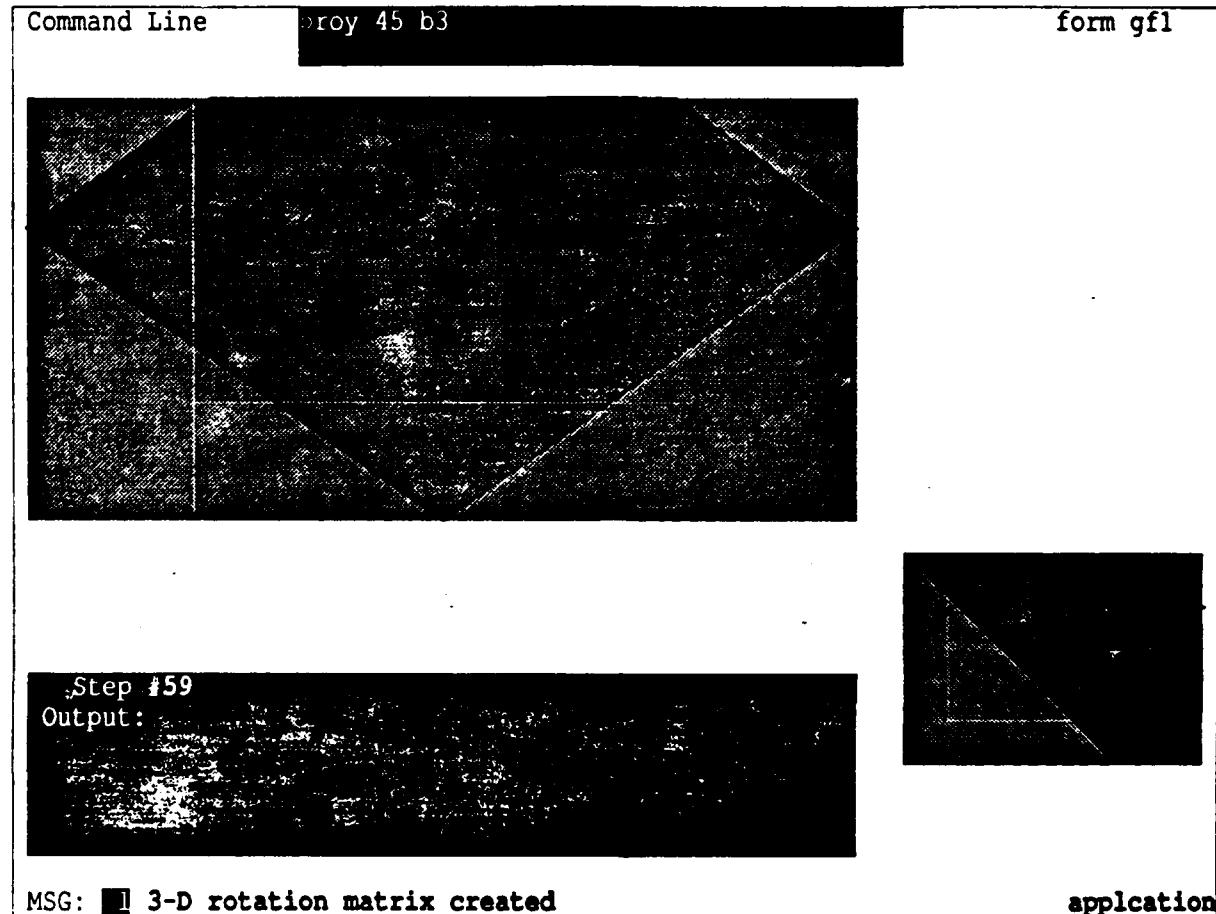
Figure 5-60b   (AFTER)
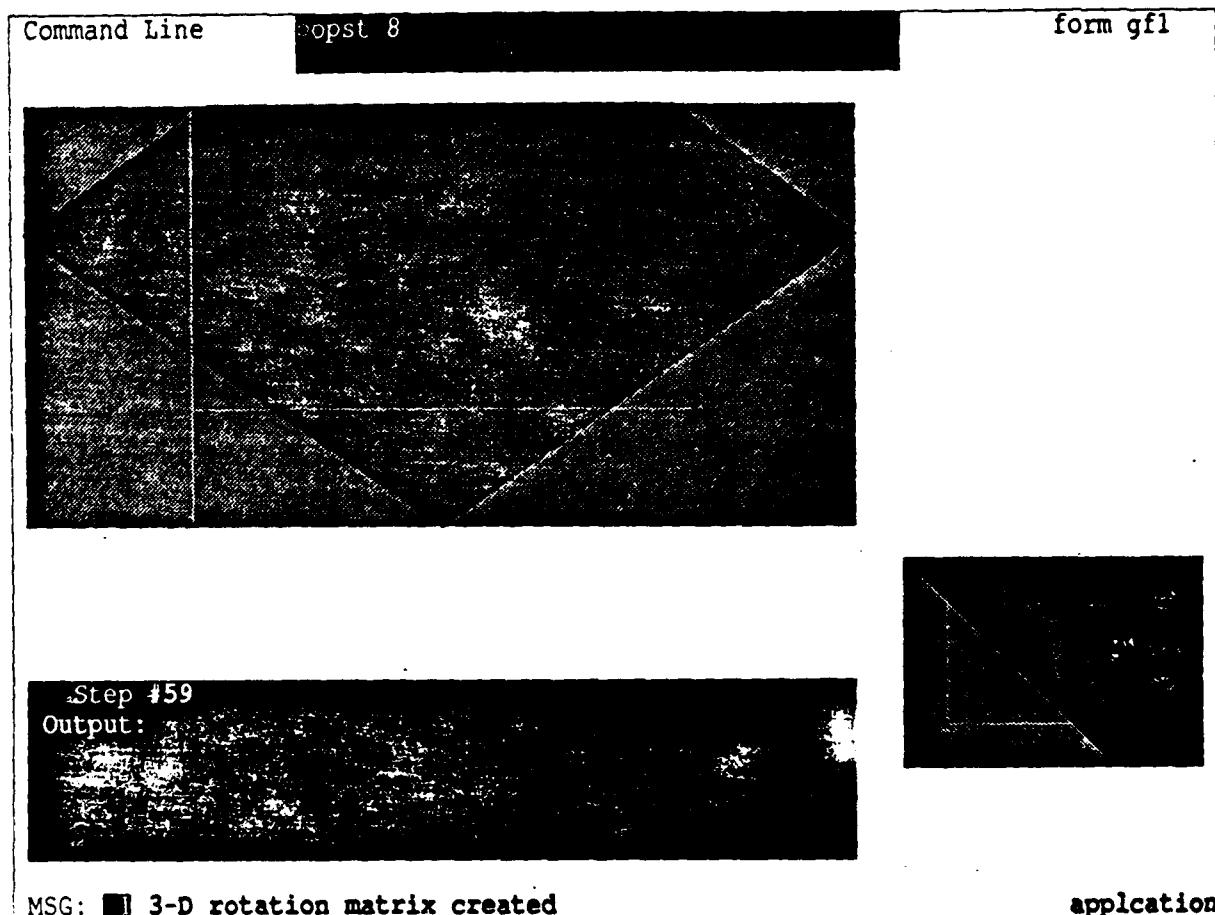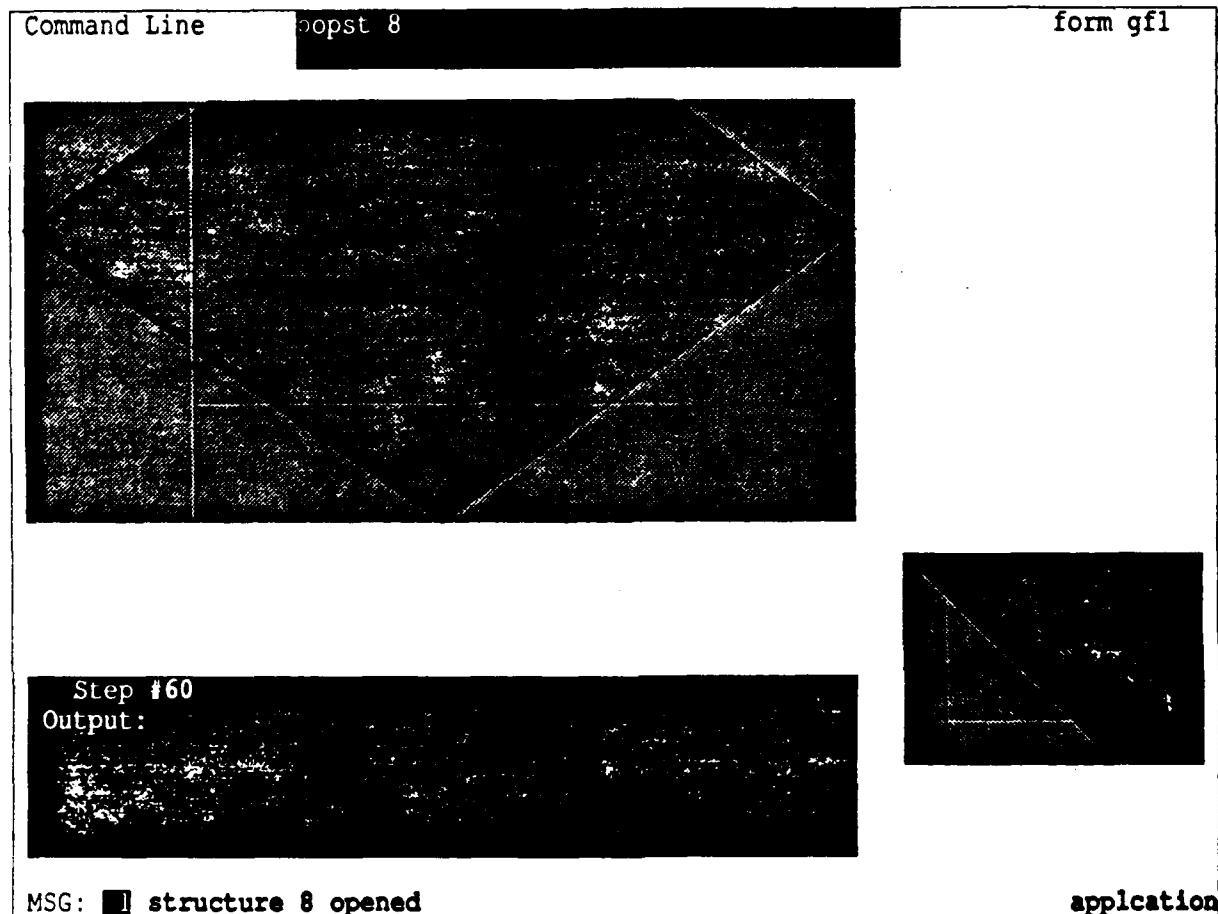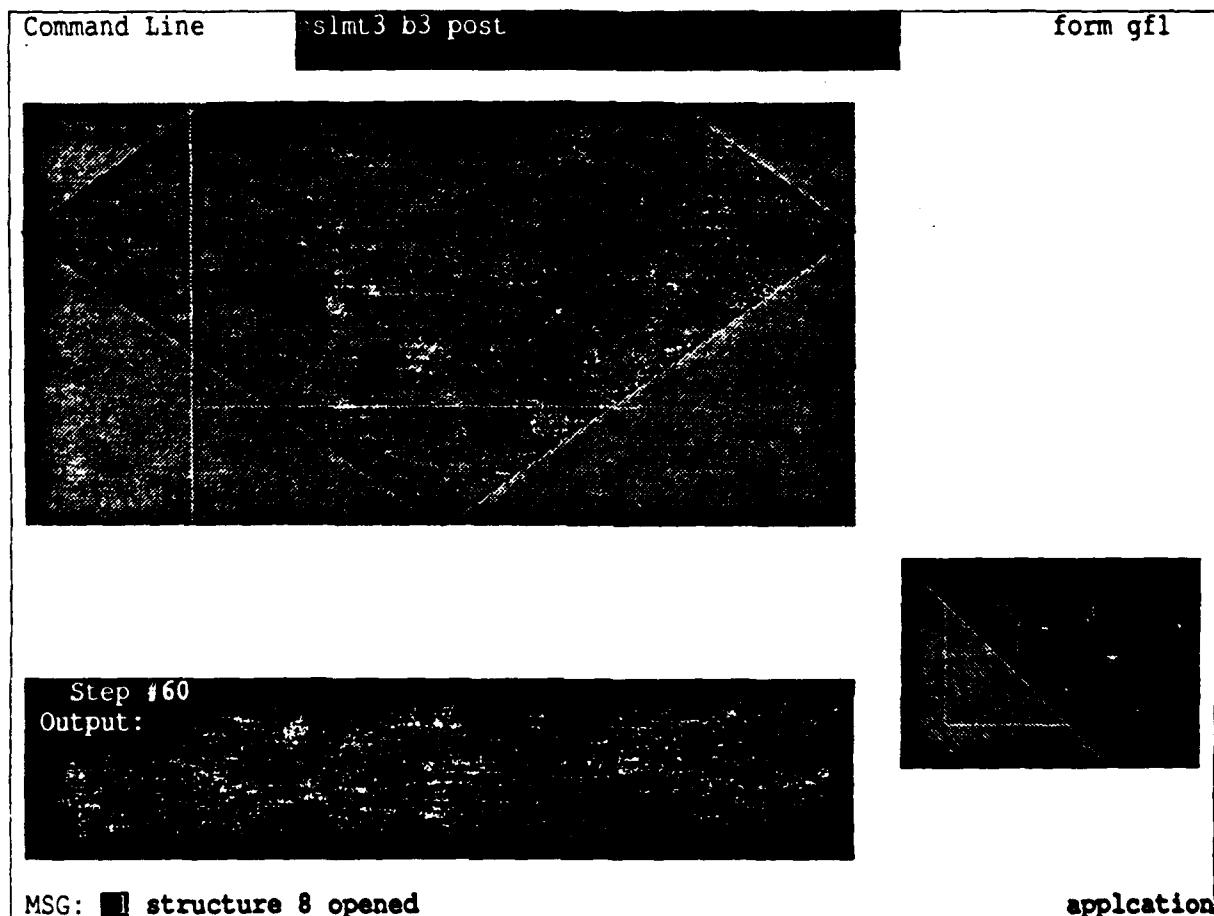
Figure 5-61a   (BEFORE)
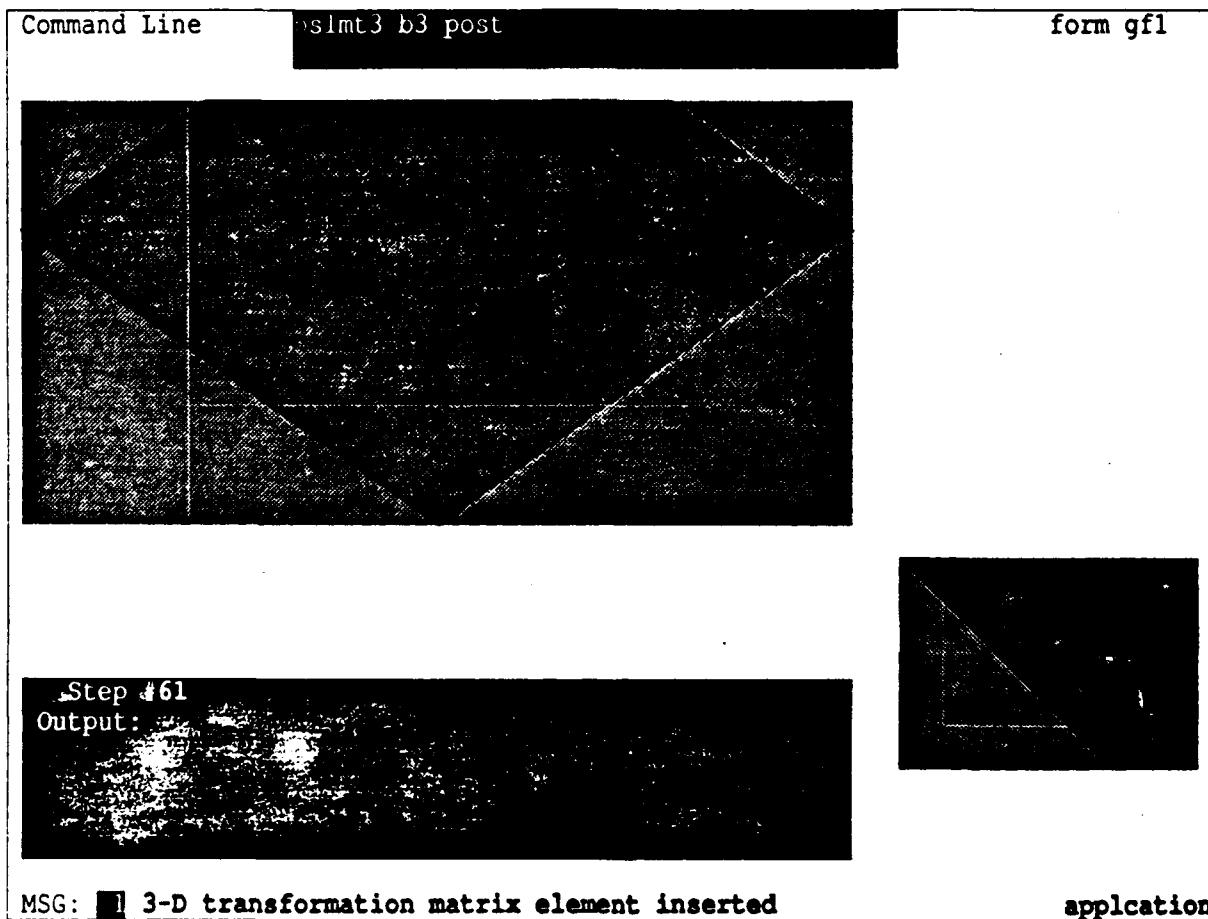
Figure 5-61b   (AFTER)

Figure 5-62a   (BEFORE)
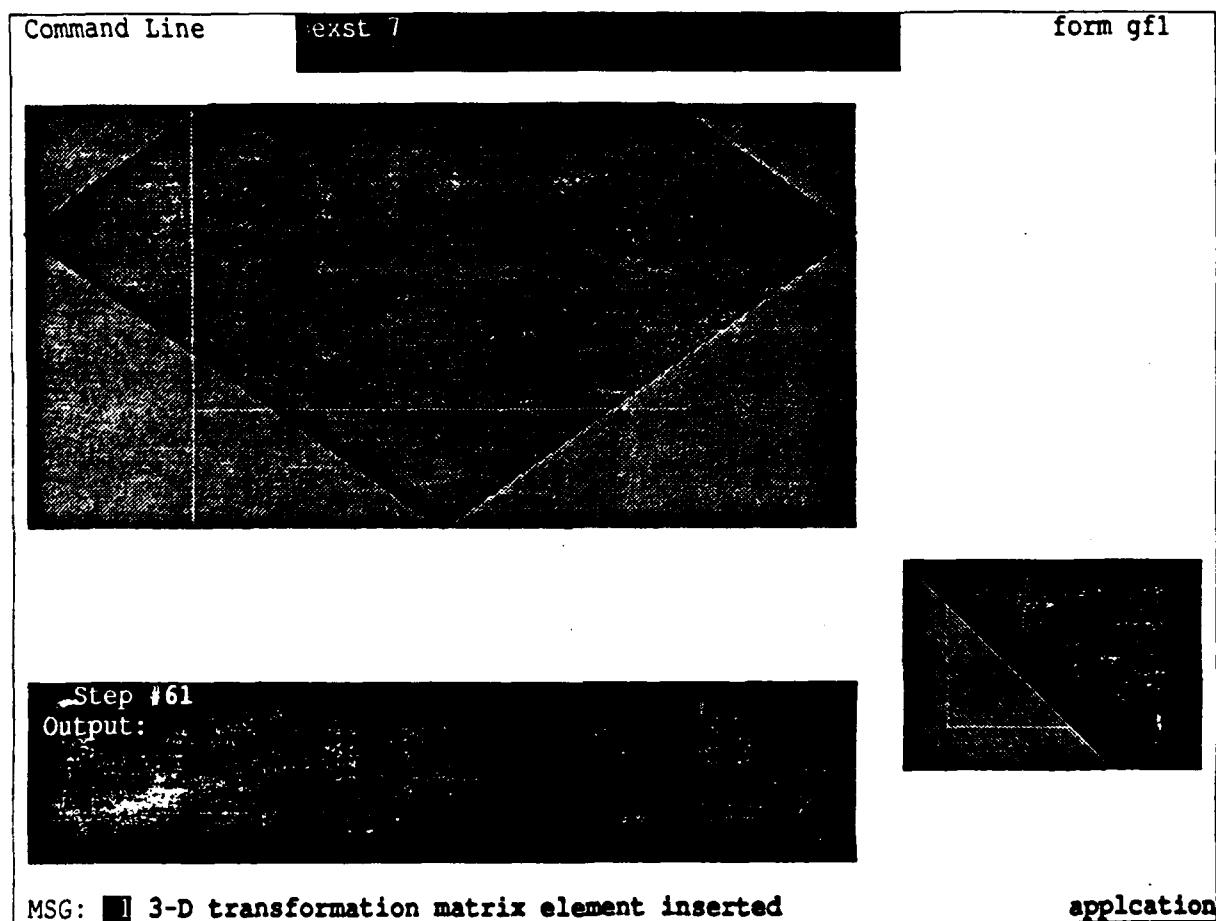
Figure 5-62b   (AFTER)
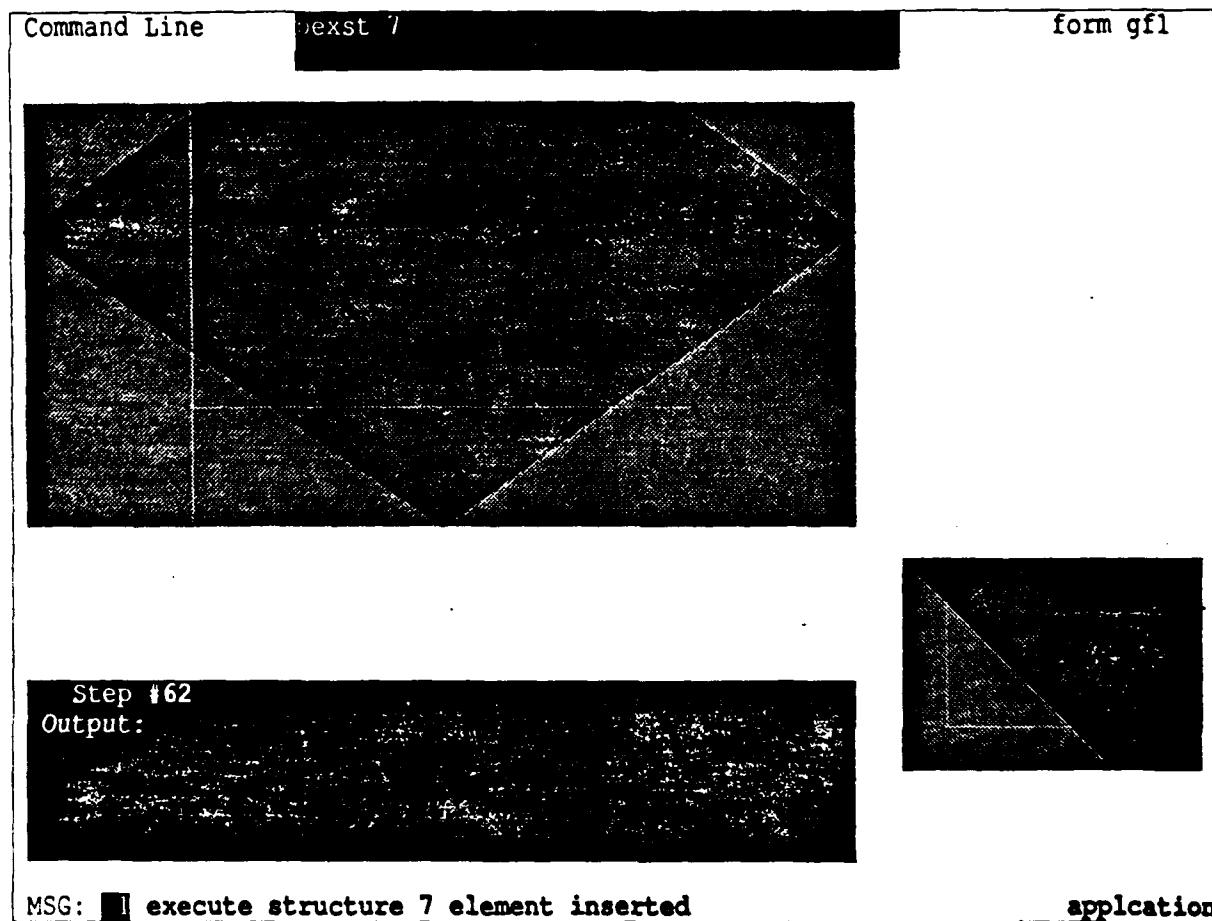
Figure 5-63a   (BEFORE)

Figure 5-63b   (AFTER)
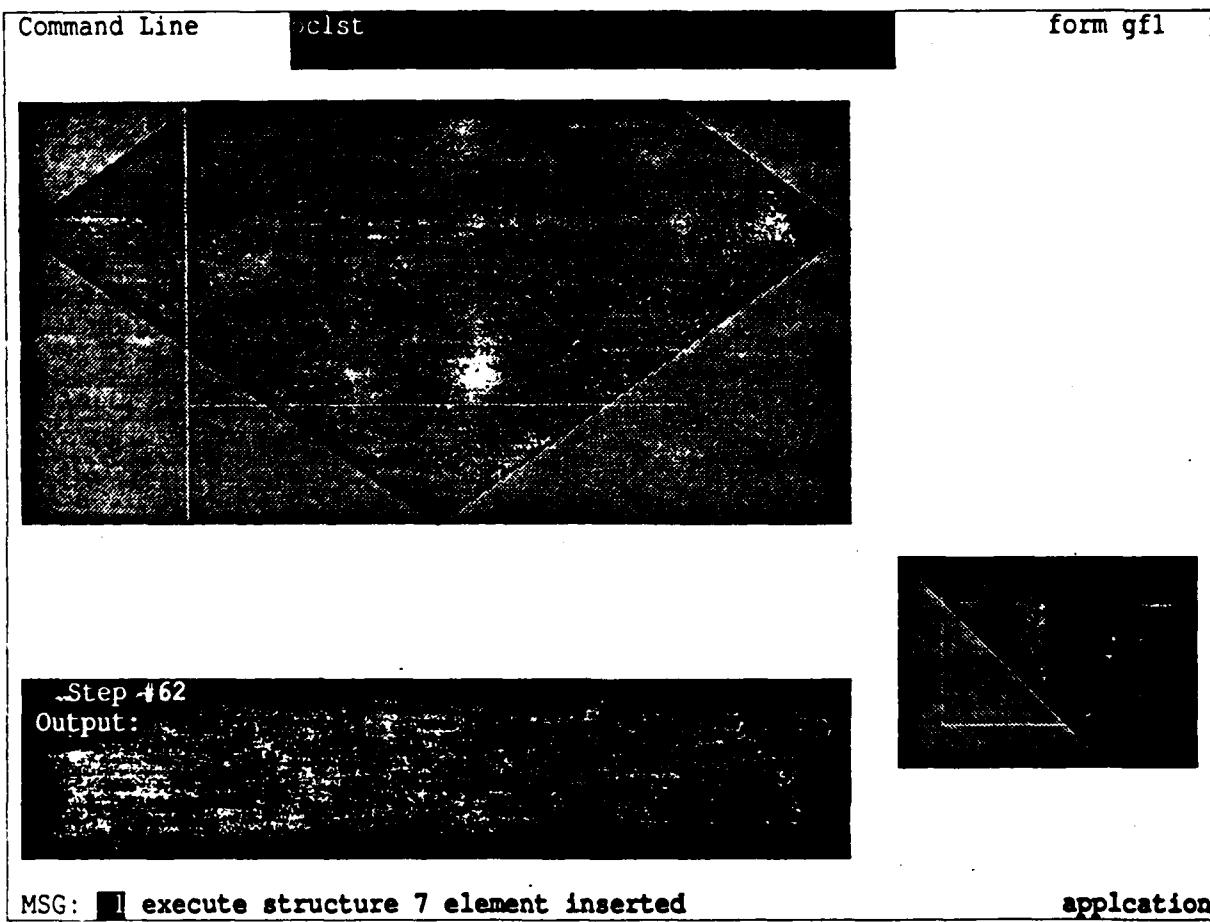
Figure 5-64a   (BEFORE)
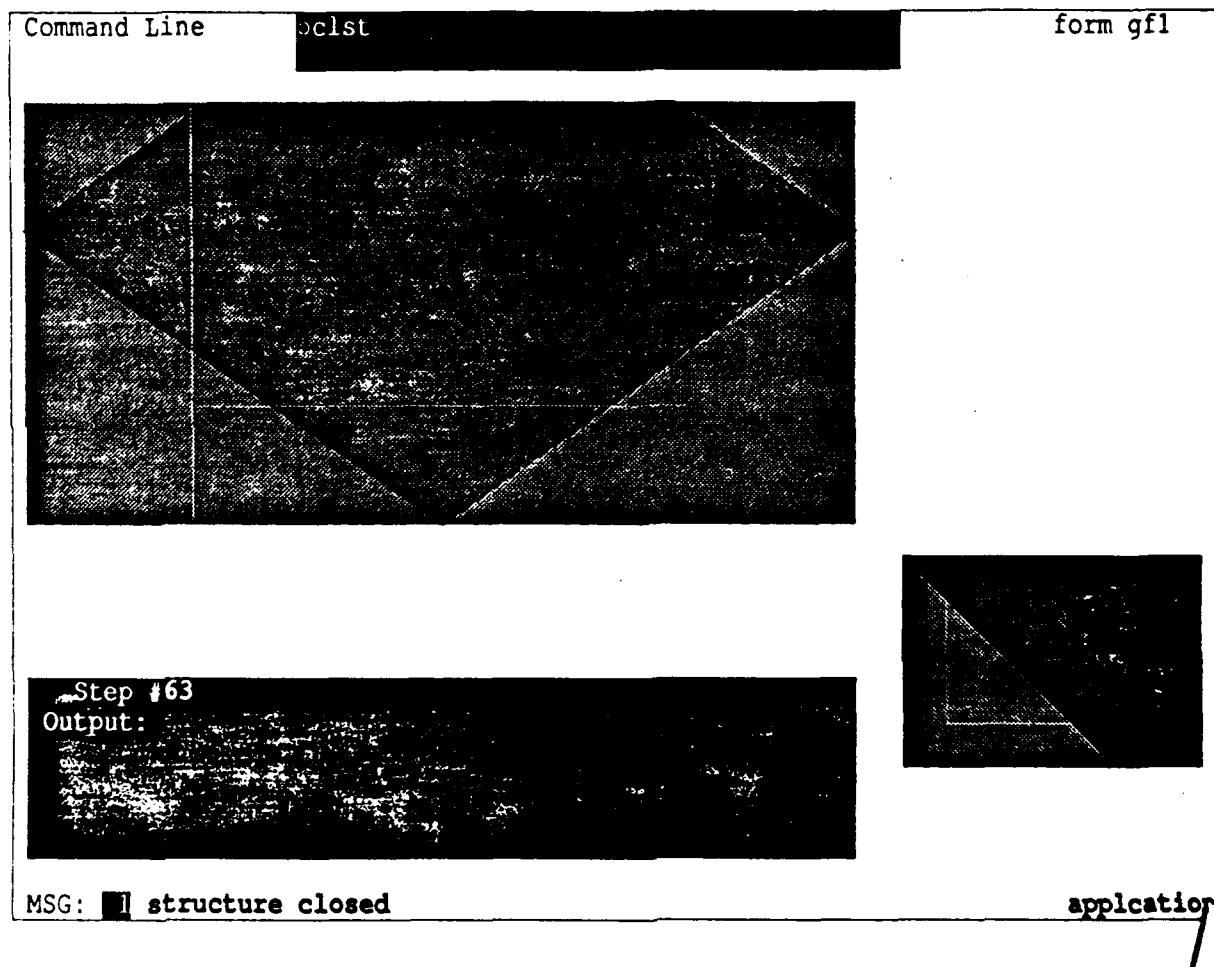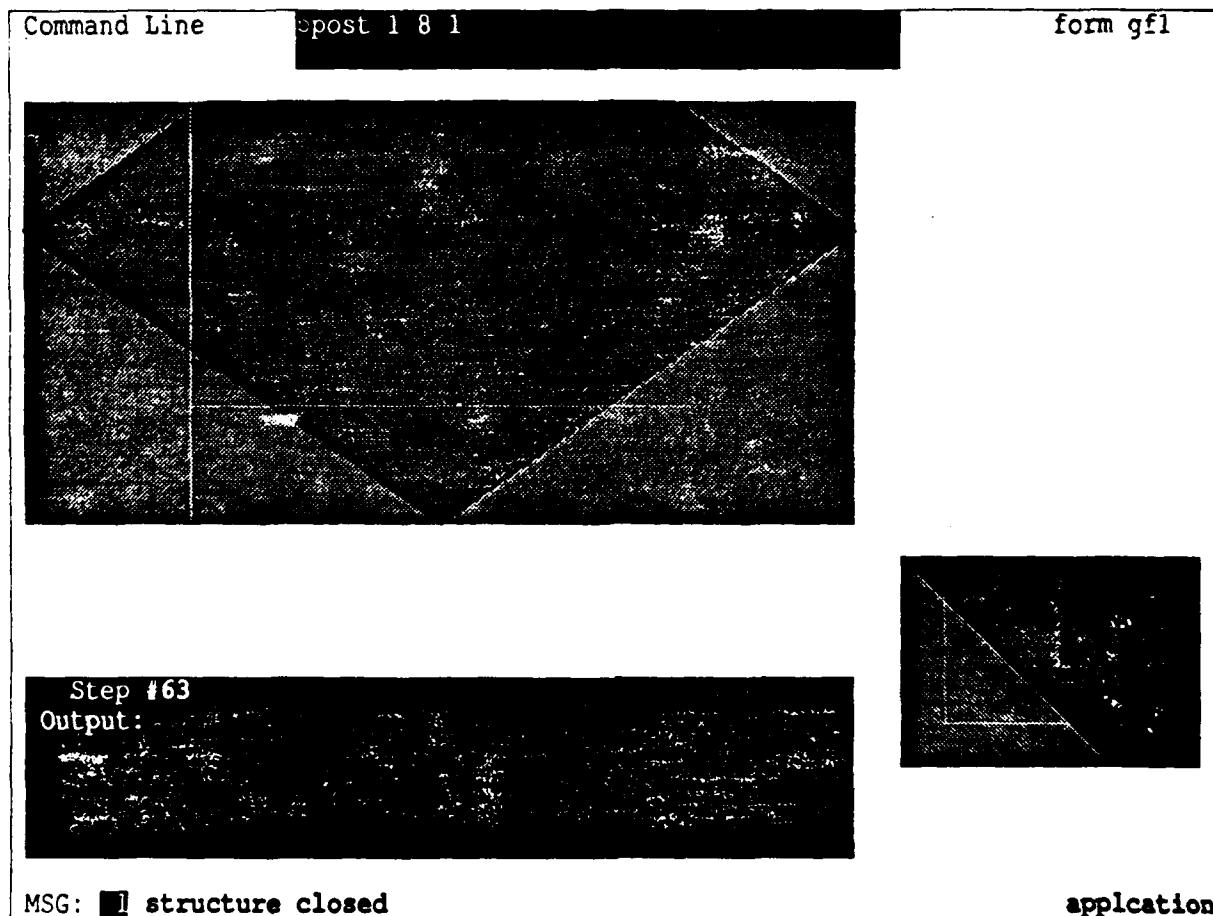
Figure 5-64b   (AFTER)

Figure 5-65a   (BEFORE)
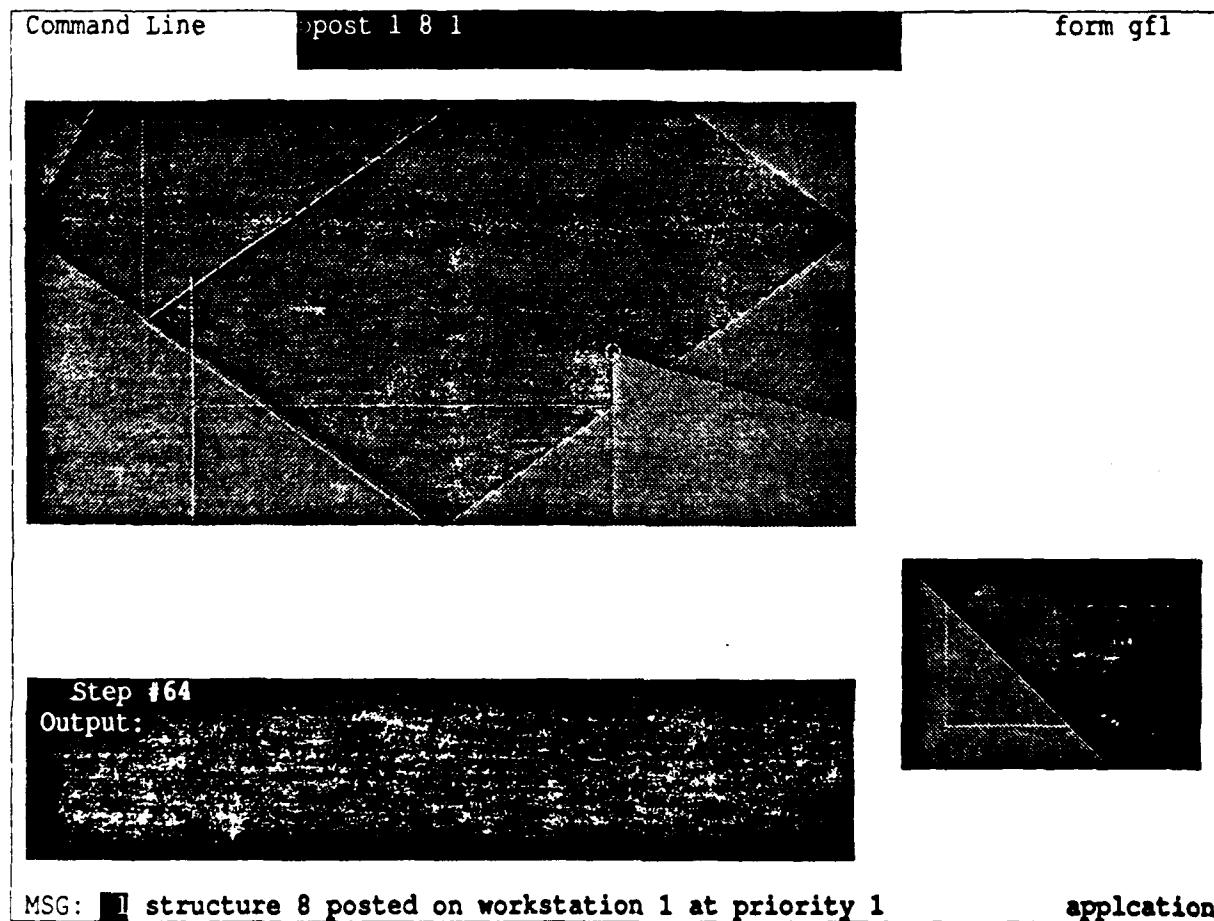
Figure 5-65b   (AFTER)

Figure 5-66a (BEFORE)

Figure 5-66b   (AFTER)

Figure 5-67a   (BEFORE)
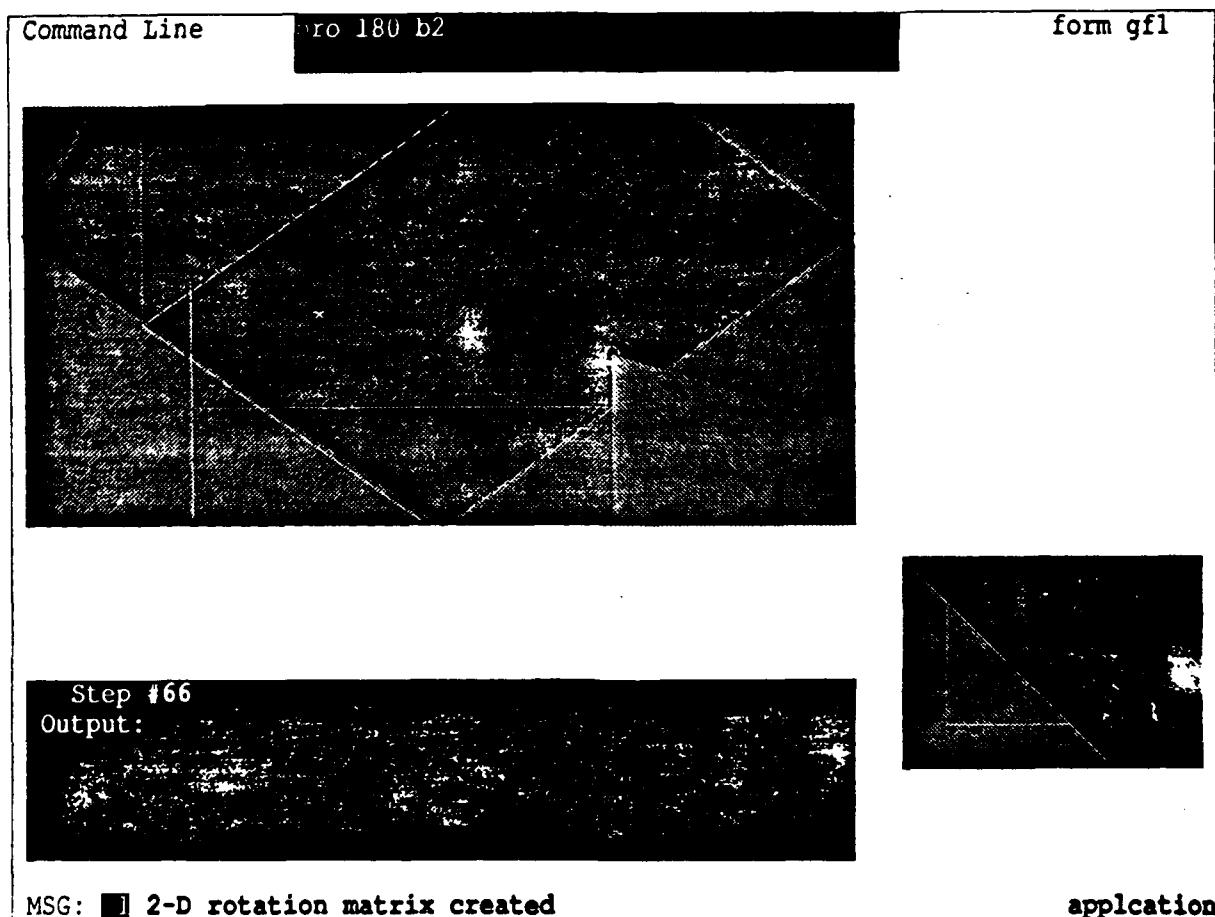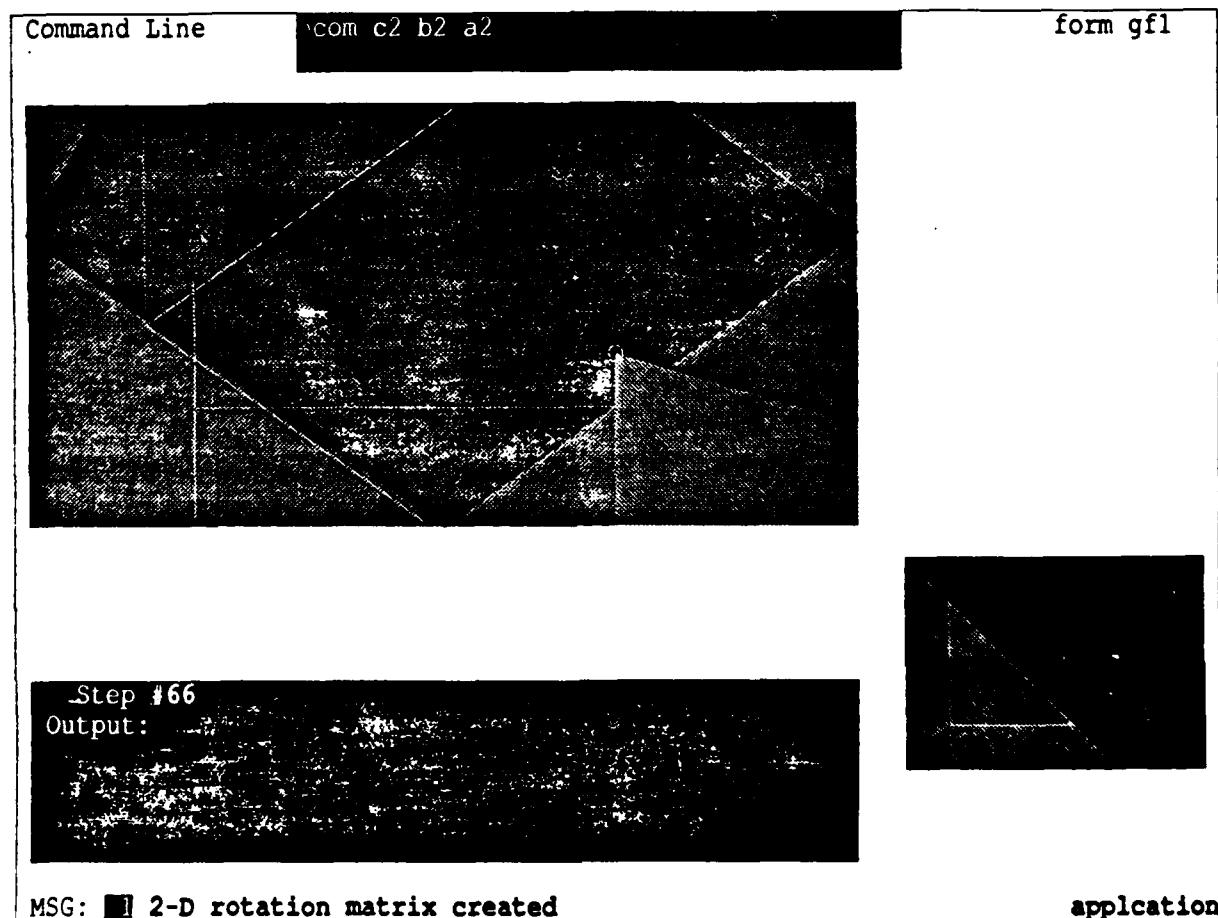
Figure 5-67b   (AFTER)

Figure 5-68a (BEFORE)
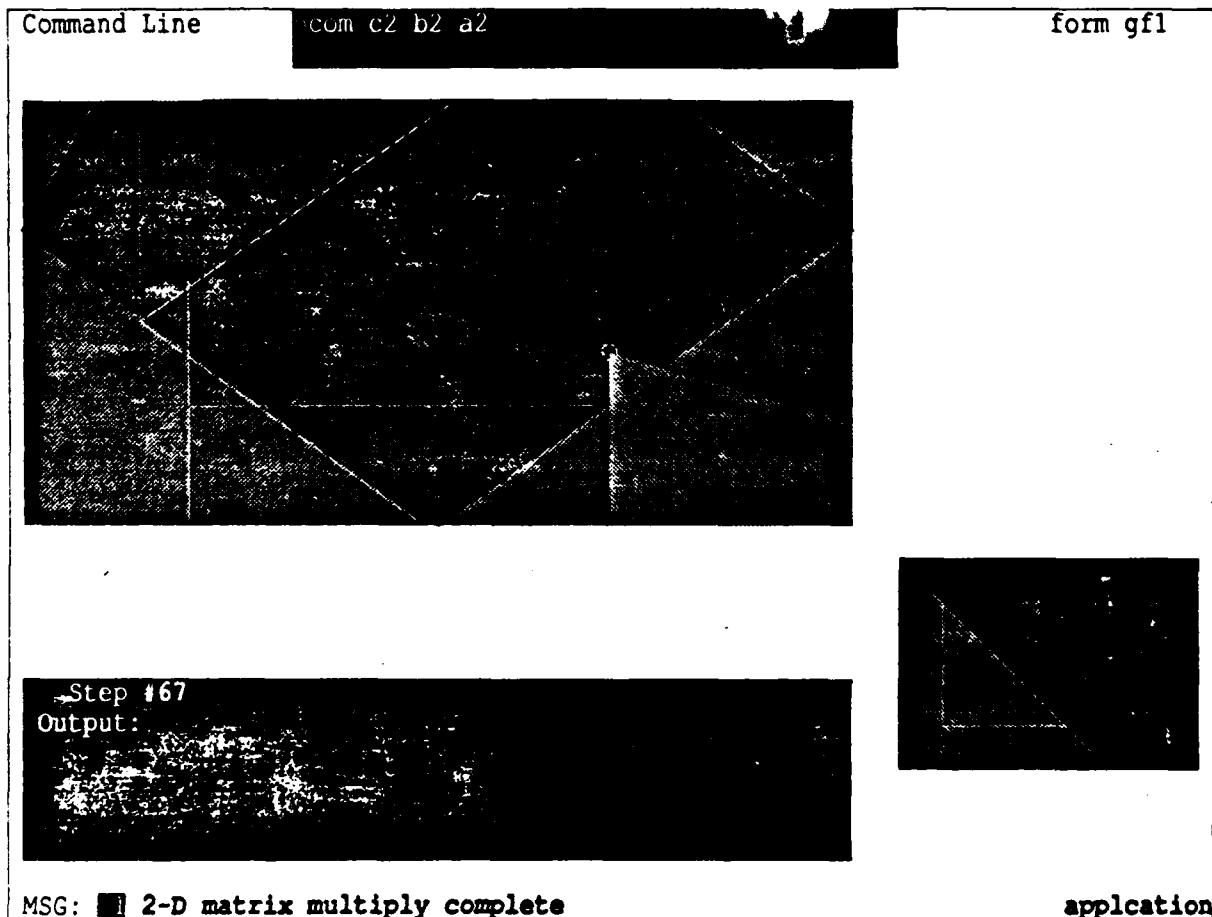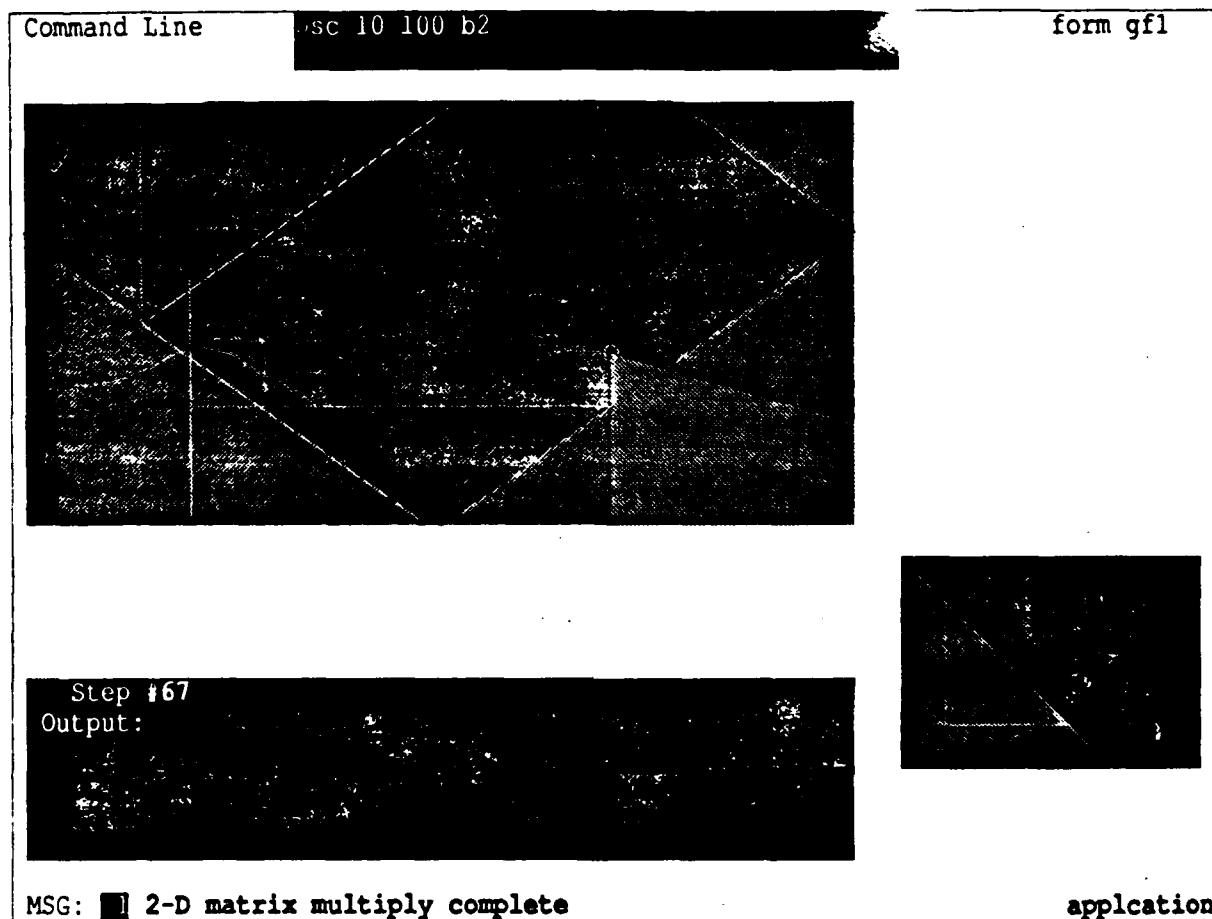
Figure 5-68b   (AFTER)

Figure 5-69a   (BEFORE)
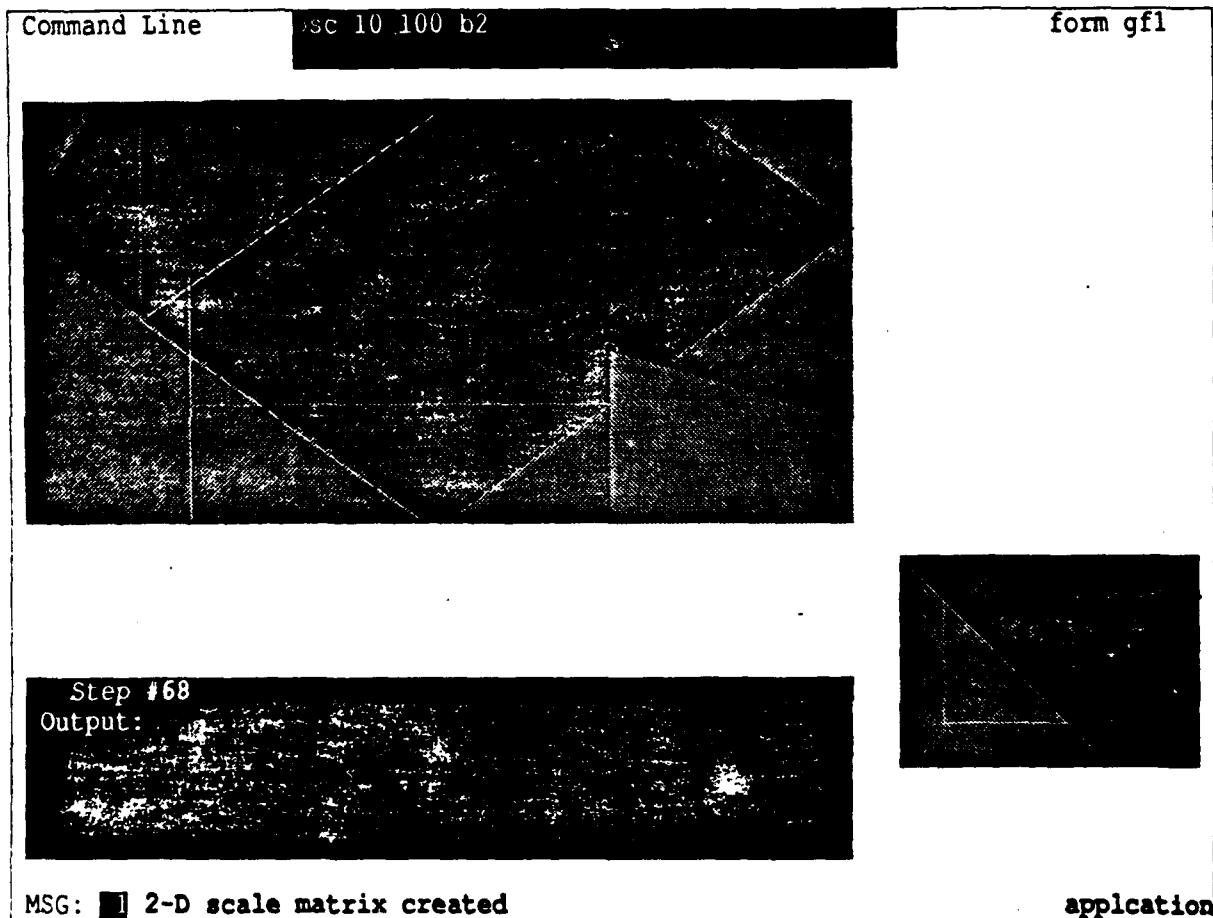
Figure 5-69b   (AFTER)
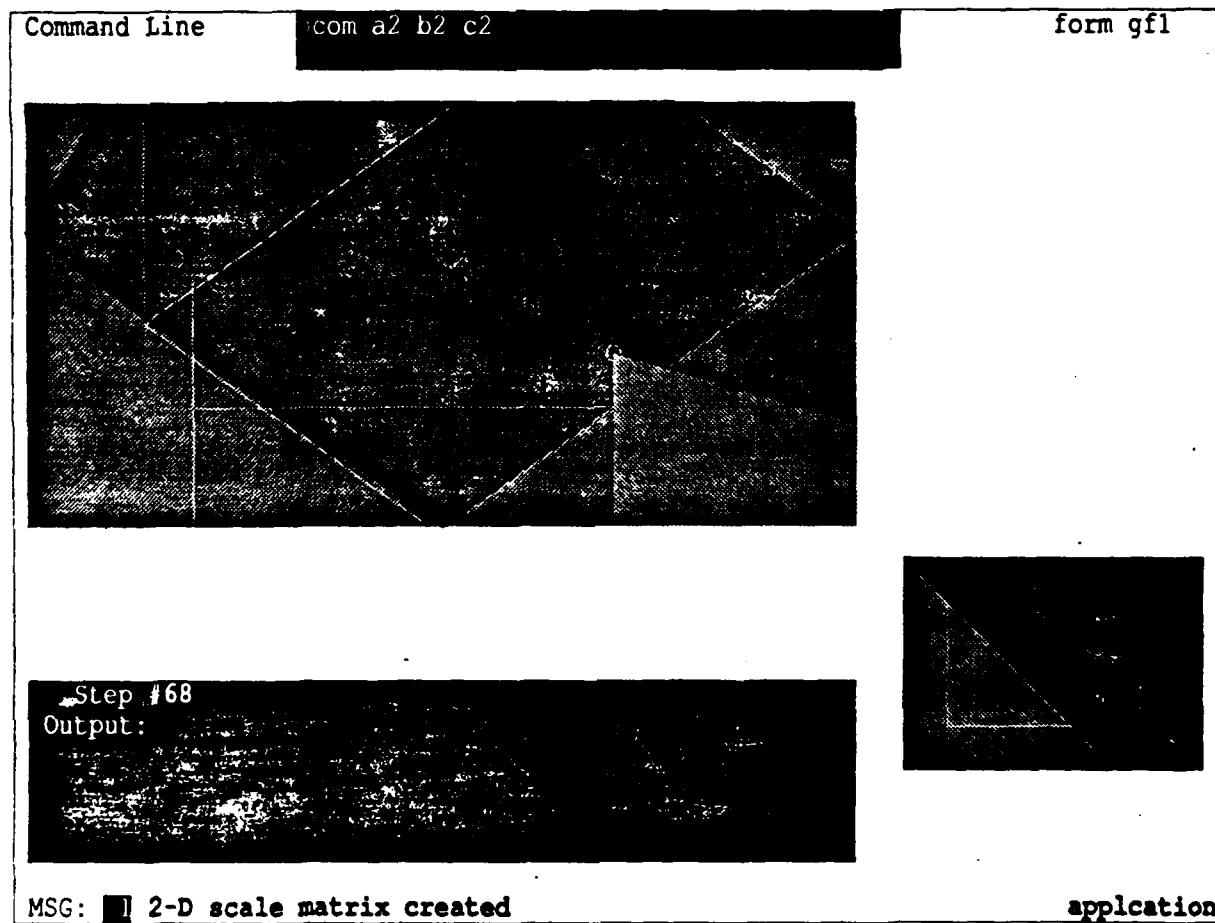
Figure 5-70a   (BEFORE)
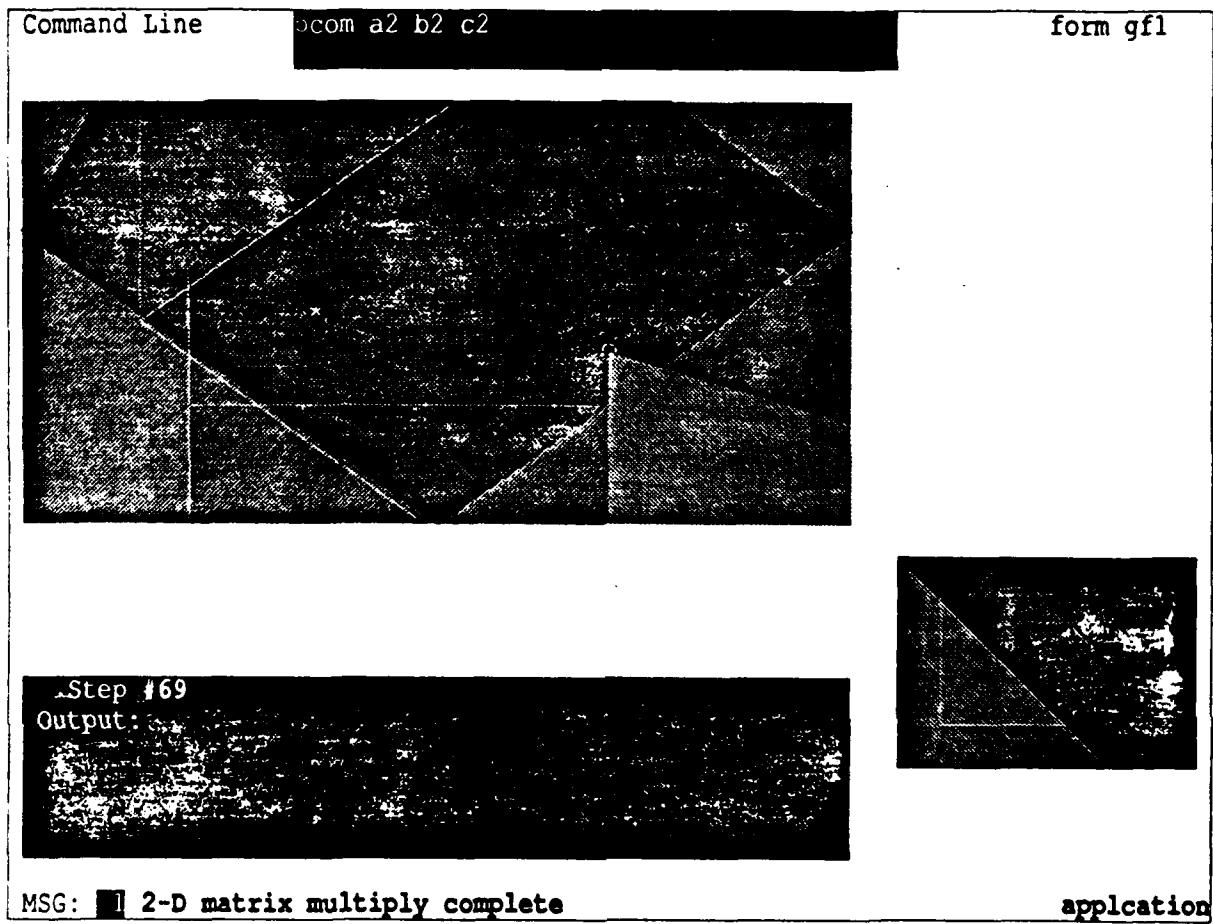
Figure 5-70b   (AFTER)

Figure 5-71a   (BEFORE)

Command Line   otp 1 0 c2   form gfl

Step #70
Output transformed coordinates= -110, -1000

MSG: 2-D transform point complete   applcation

Figure 5-71b   (AFTER)

Command Line   )tp 0 1 c2   form gf1

_Step #70
Output: Transformed coordinates= -110, -1000
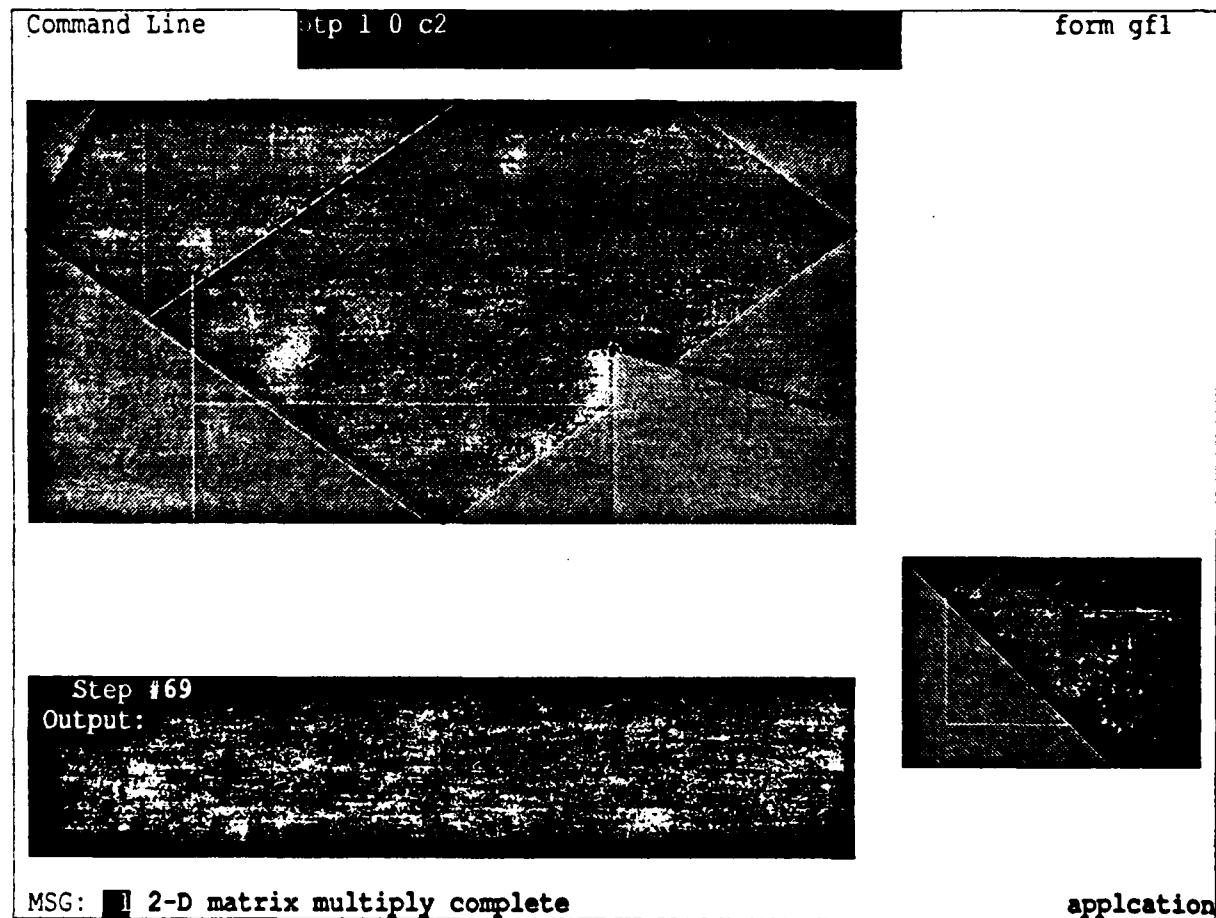
MSG: ▊ 2-D transform point complete   applcation
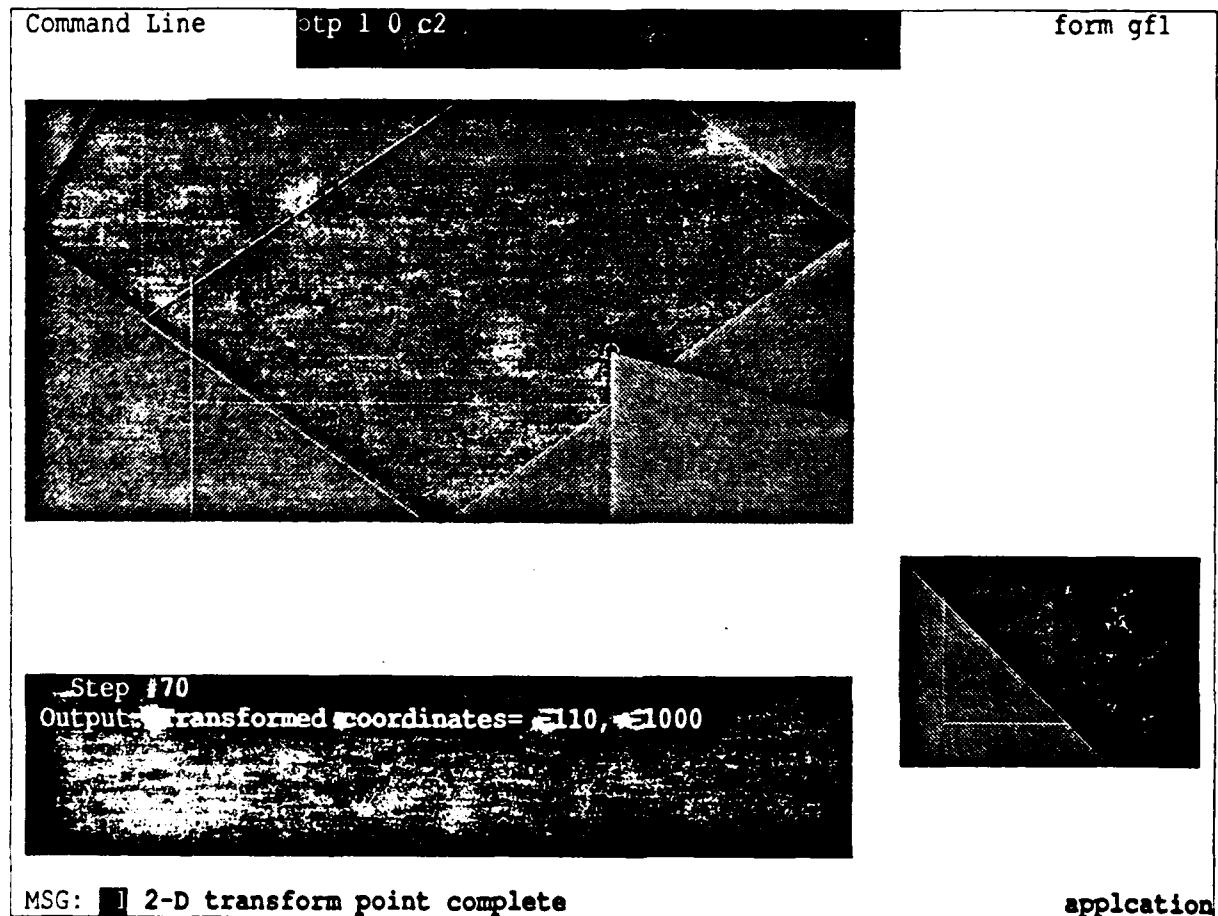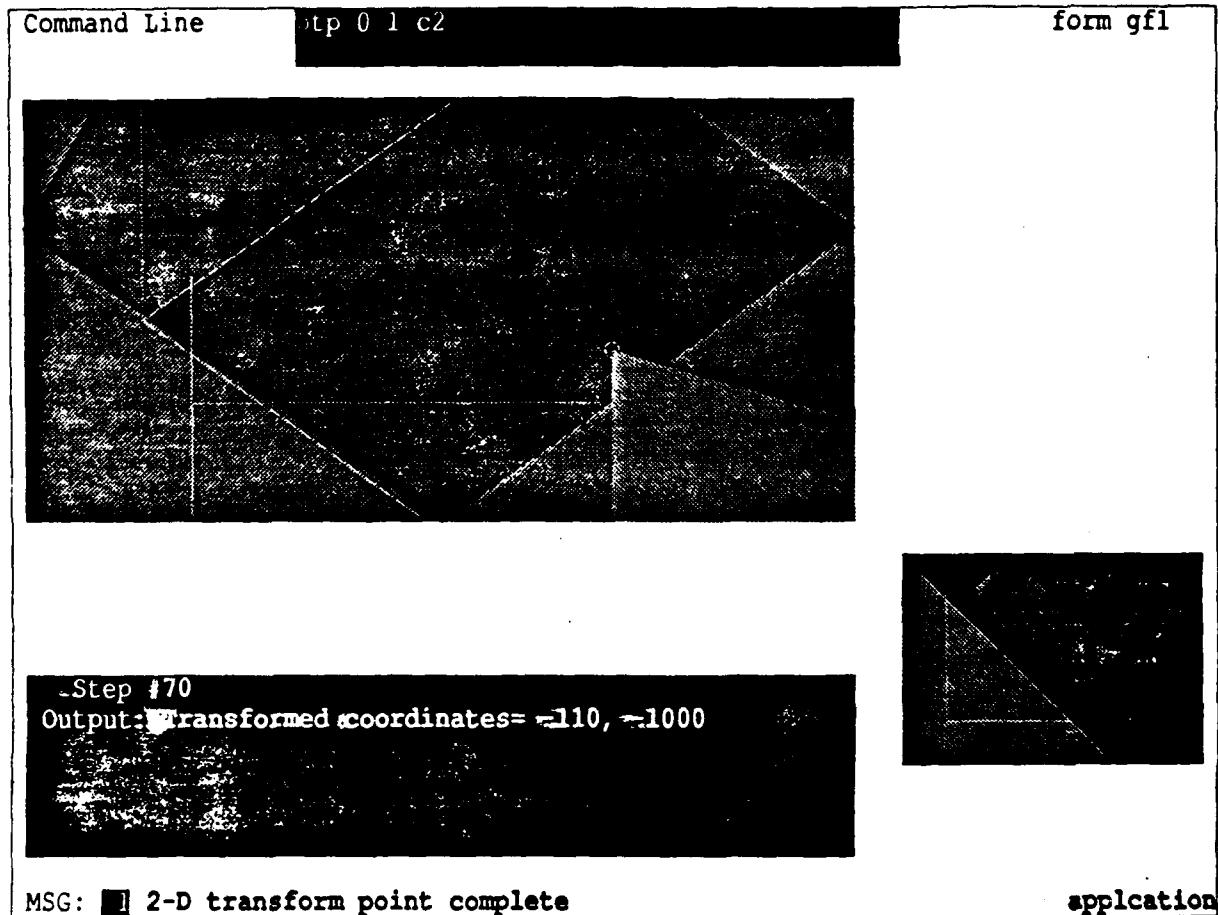
Figure 5-72a   (BEFORE)

Command Line   ɔtp 0 1 c2   form gfl
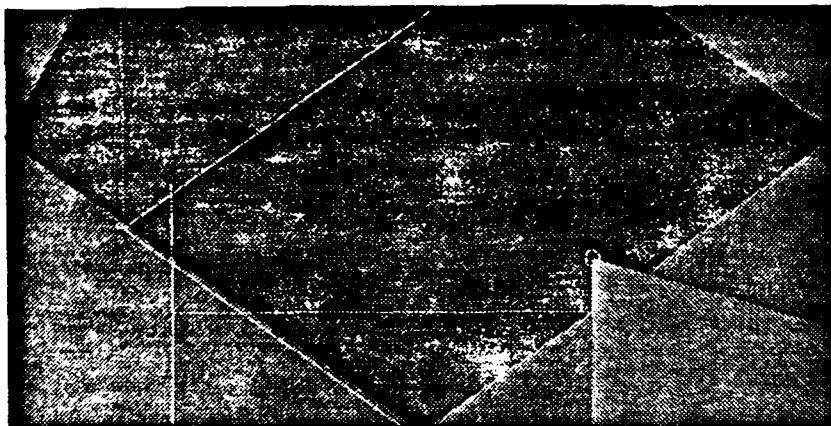
Step #71
Output: Transformed coordinates= 100, 1100

MSG: 2-D transform point complete   applcation
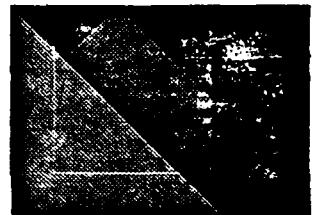
Figure 5-72b   (AFTER)
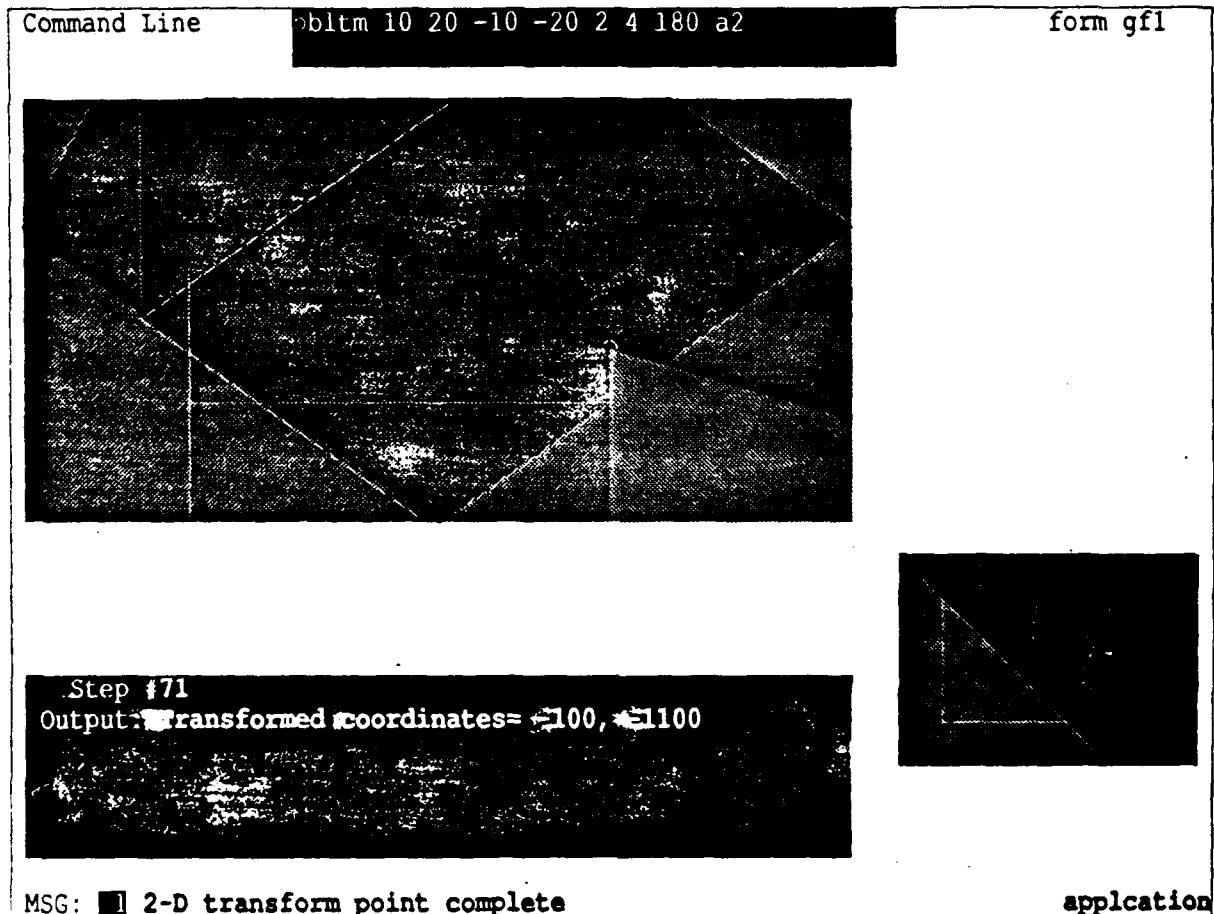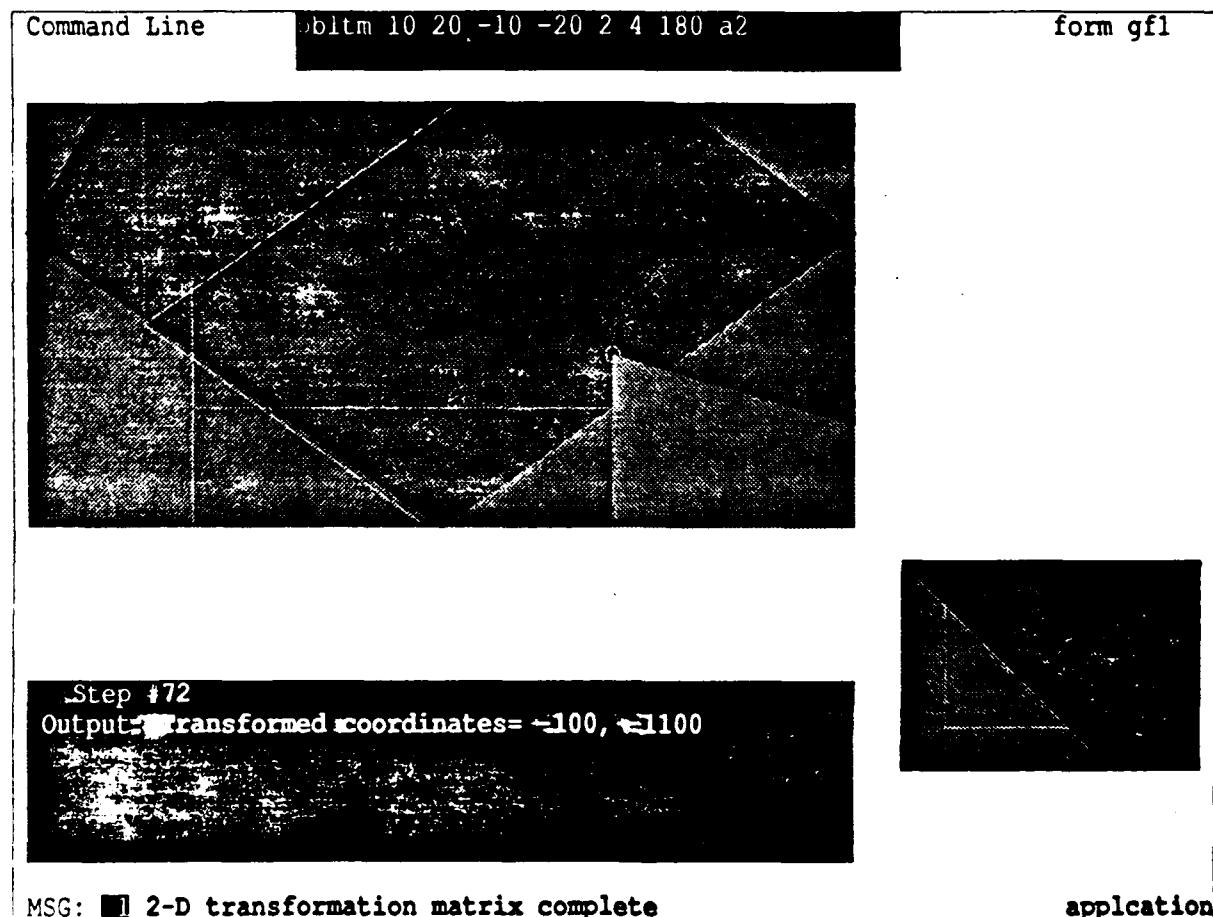
Figure 5-73a   (BEFORE)

Figure 5-73b   (AFTER)
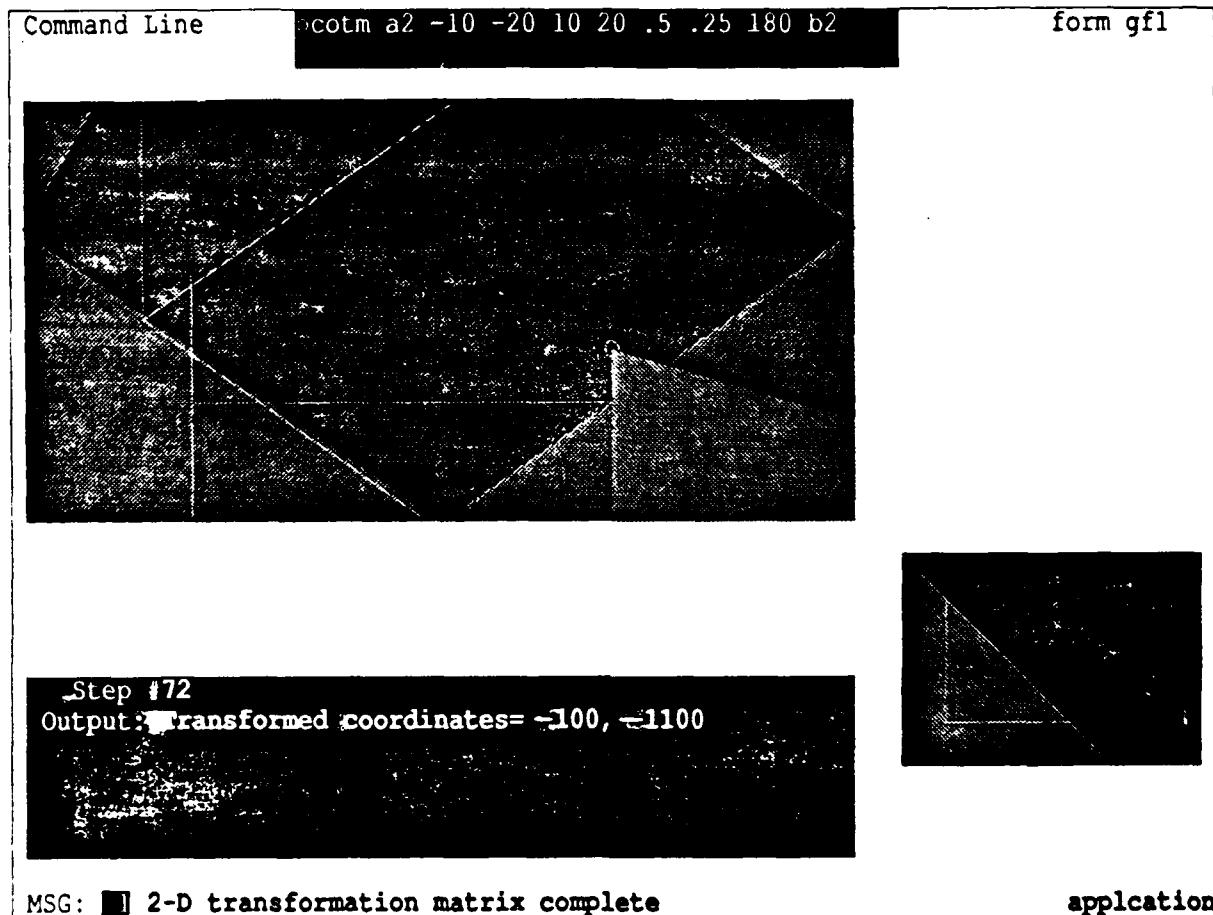
Figure 5-74a   (BEFORE)

Figure 5-74b   (AFTER)

Figure 5-/75a   (BEFORE)

Figure 5-75b   (AFTER)

Figure 5-76a   (BEFORE)

Figure 5-76b   (AFTER)

Figure 5-77a   (BEFORE)

Figure 5-77b   (AFTER)

Figure 5-78a   (BEFORE)

Figure 5-78b (AFTER)

Figure 5-79a   (BEFORE)

Figure 5-79b   (AFTER)

Figure 5-80a   (BEFORE)

Command Line   )com3 a3 b3 c3   form gfl
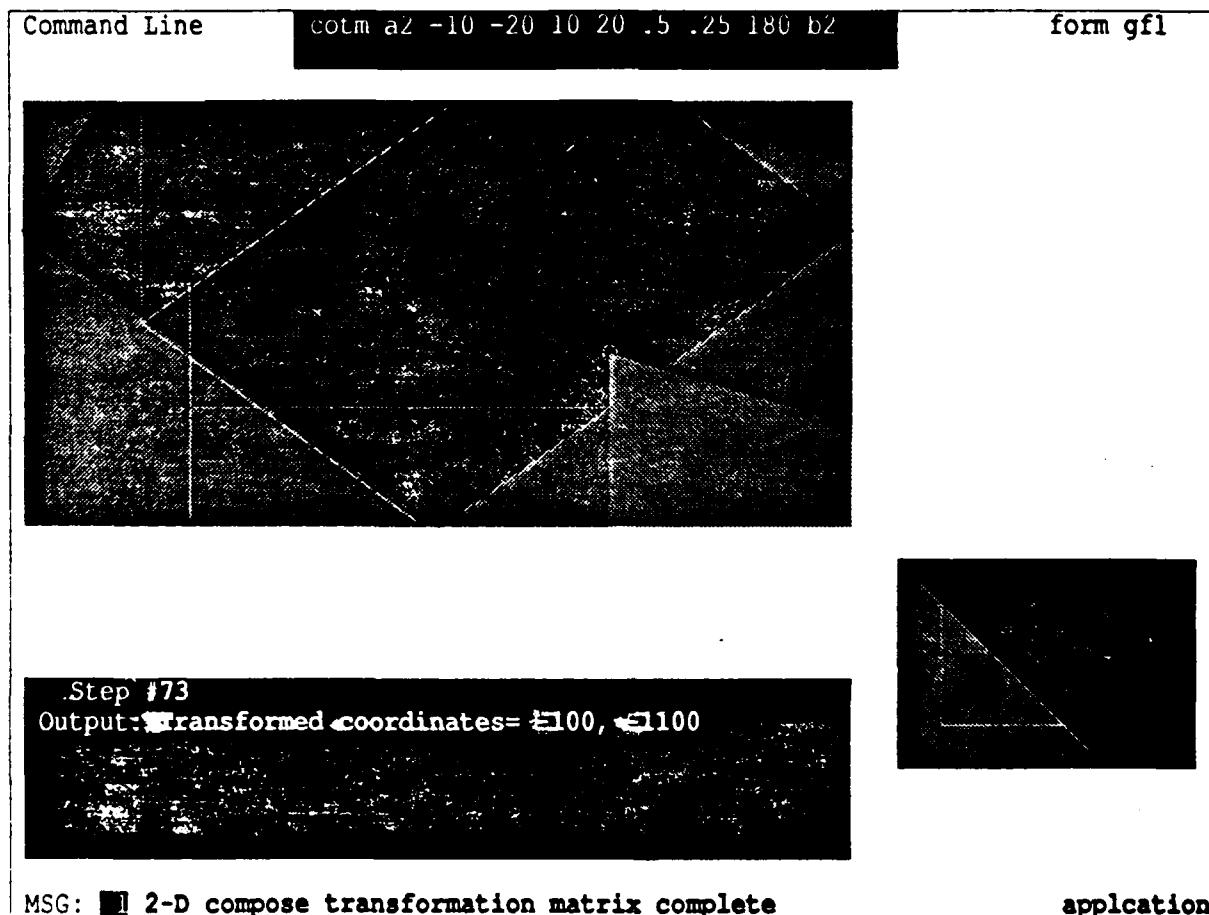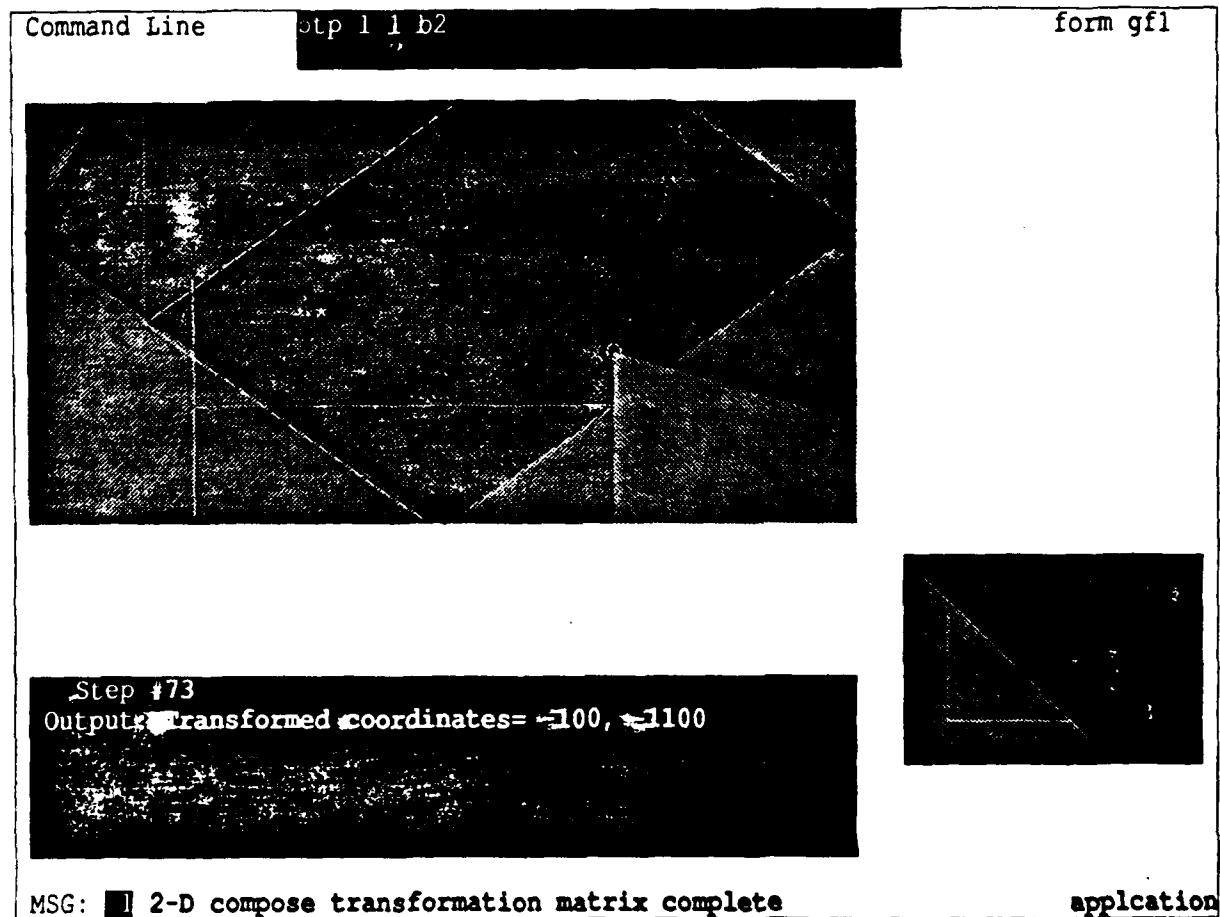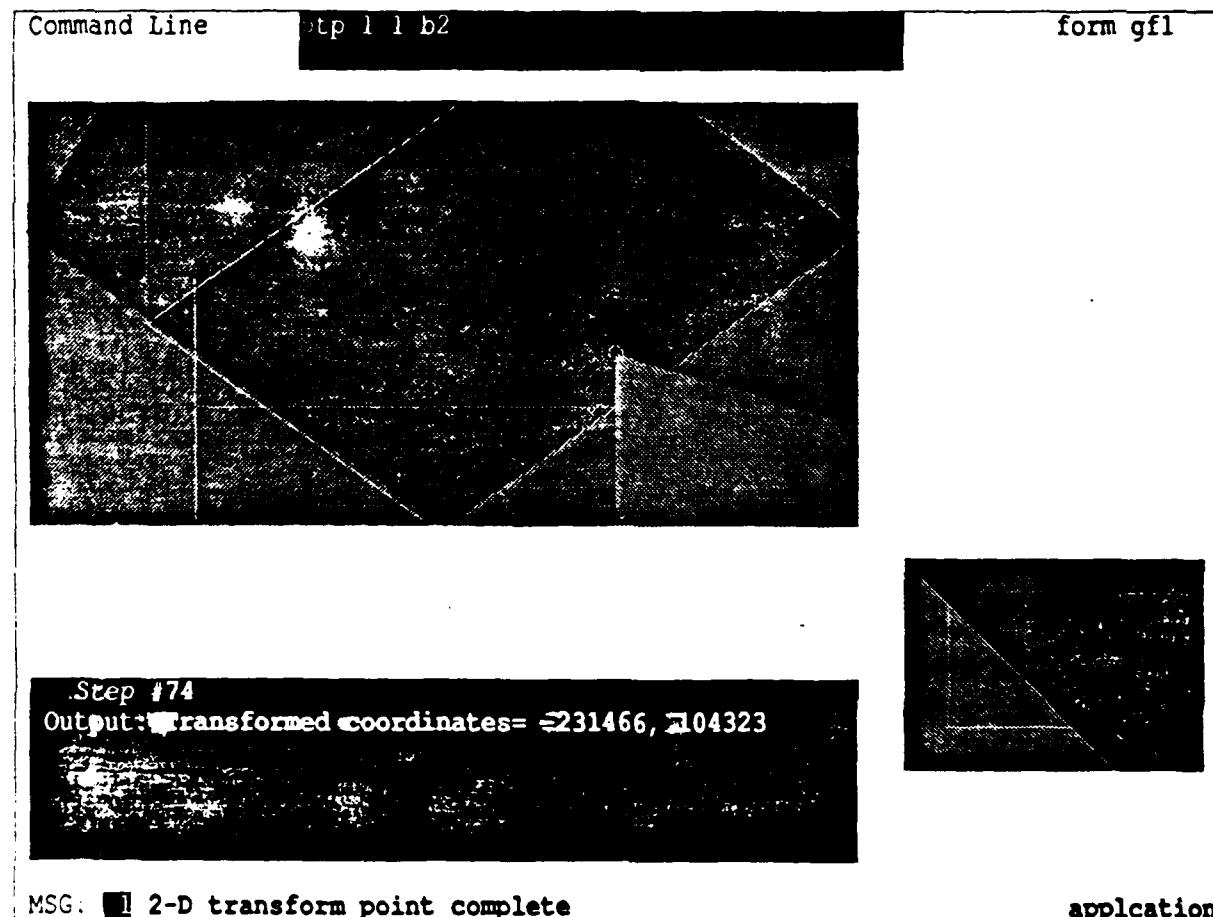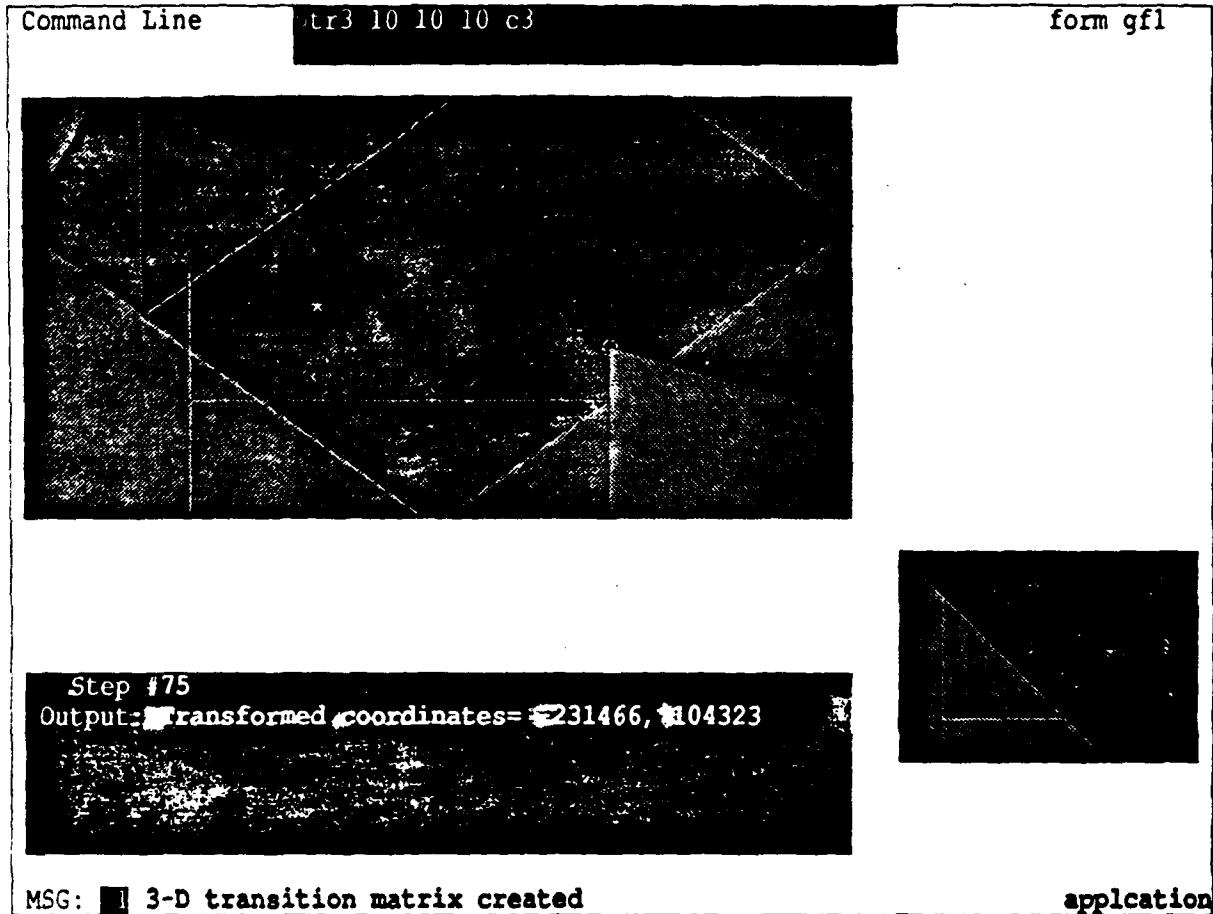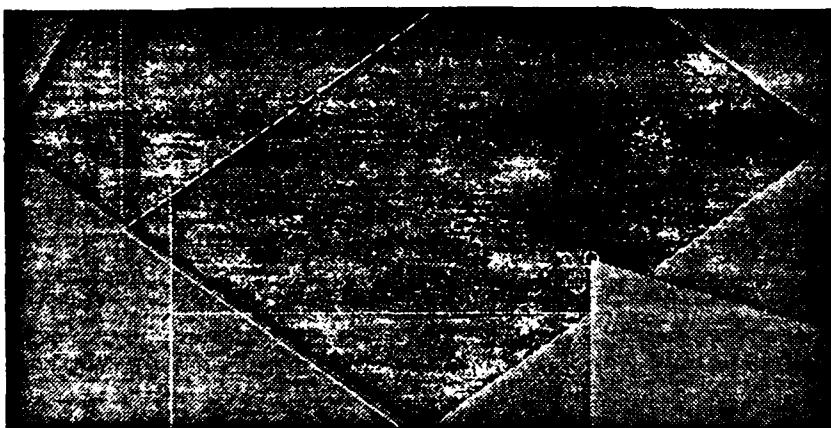
Step #79
Output: transformed coordinates= -231466, 104323

MSG: ▊ 3-D matrix multiply complete   applcation
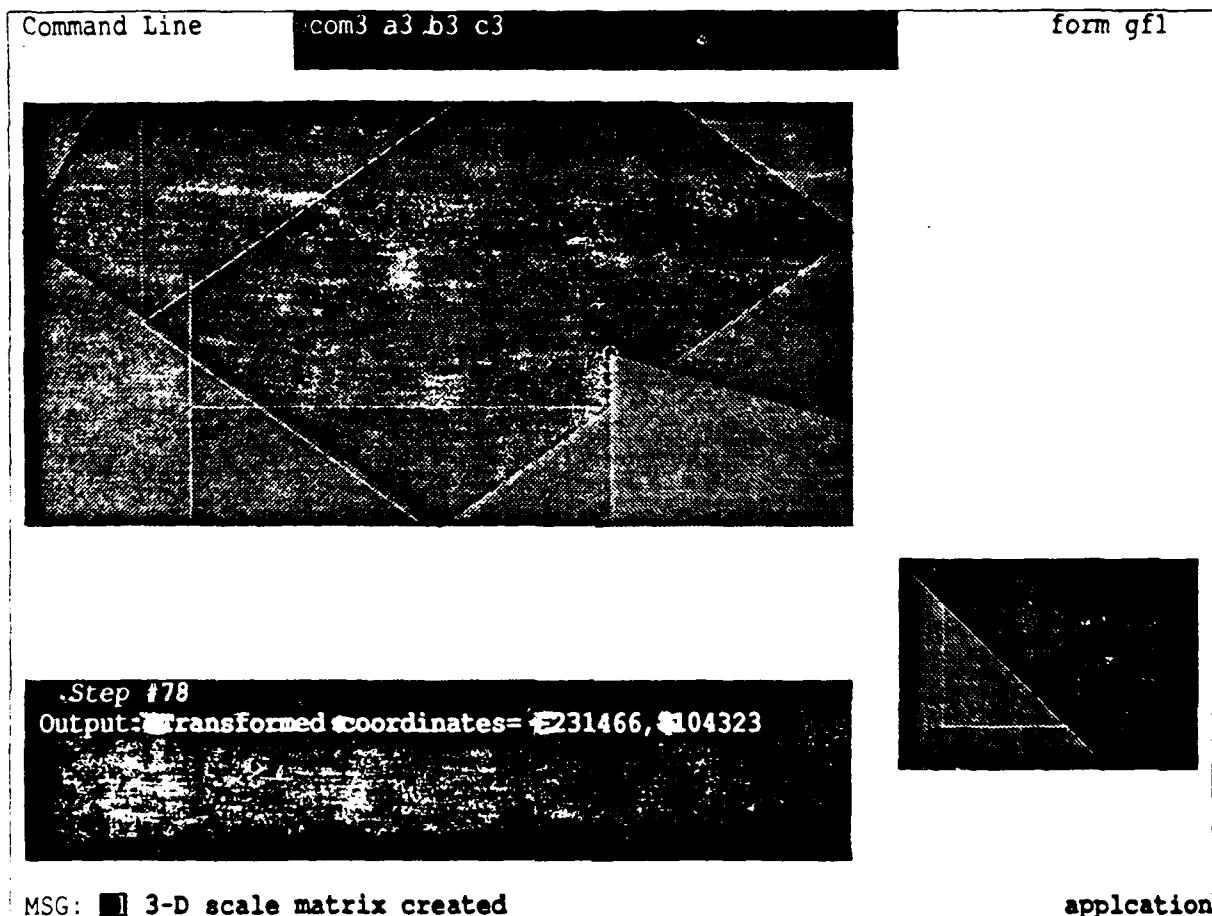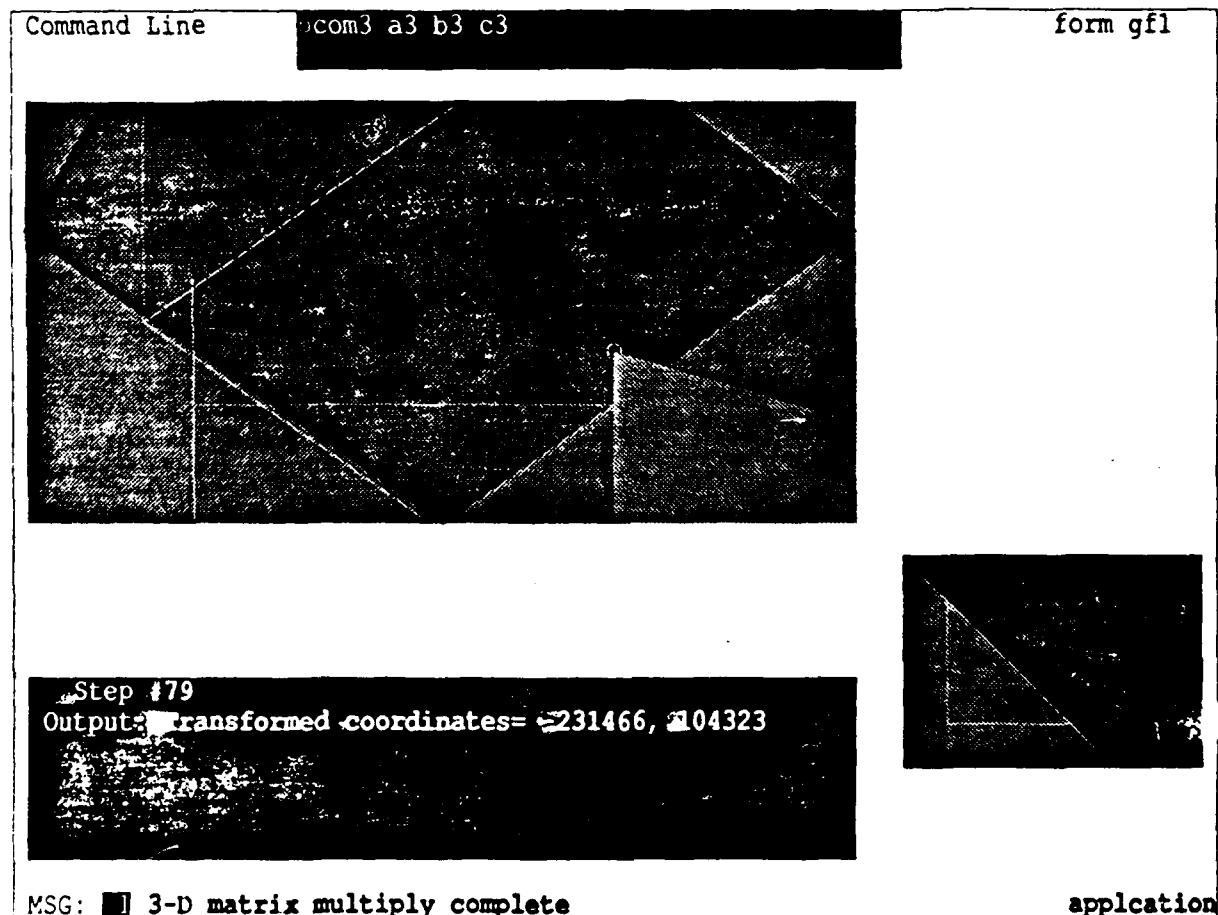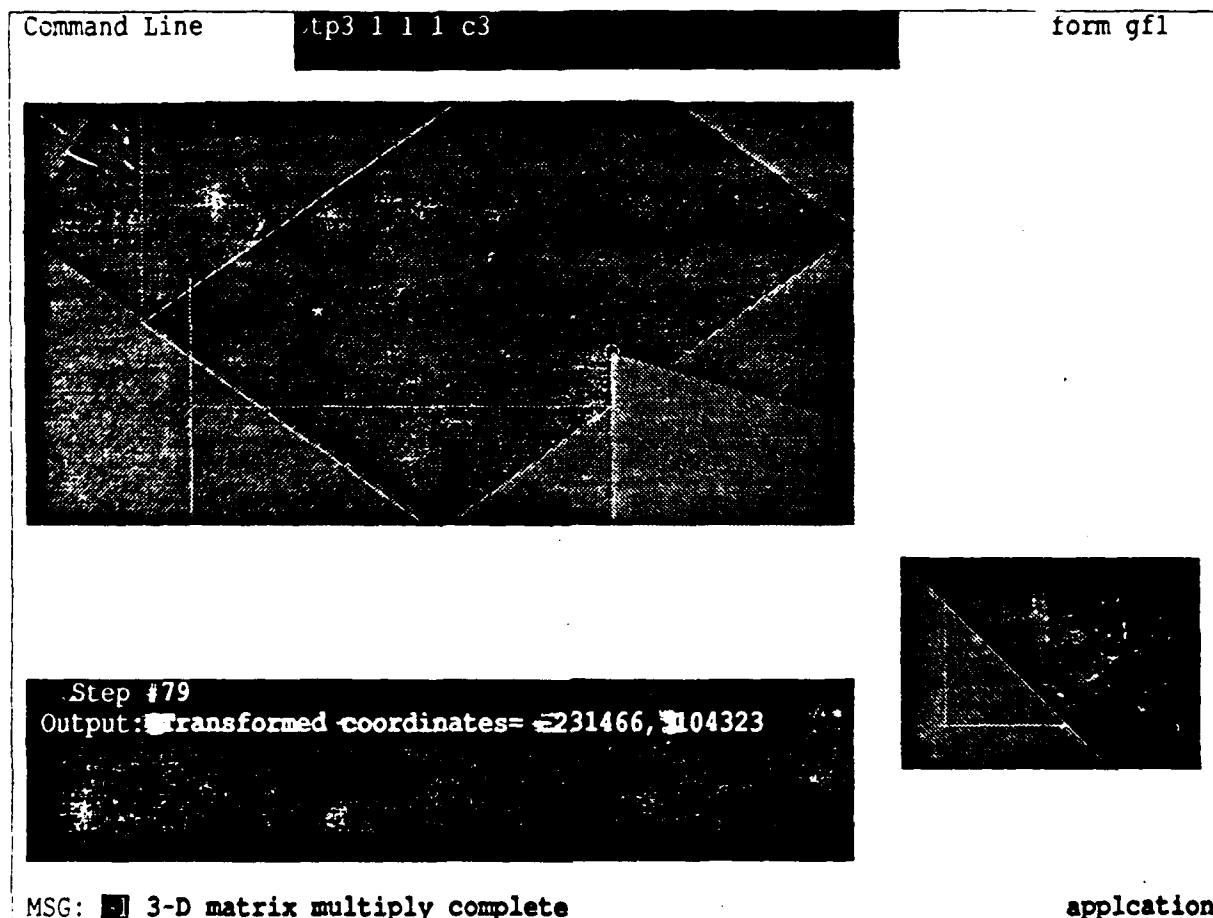
Figure 5-80b   (AFTER)

Figure 5-81a   (BEFORE)
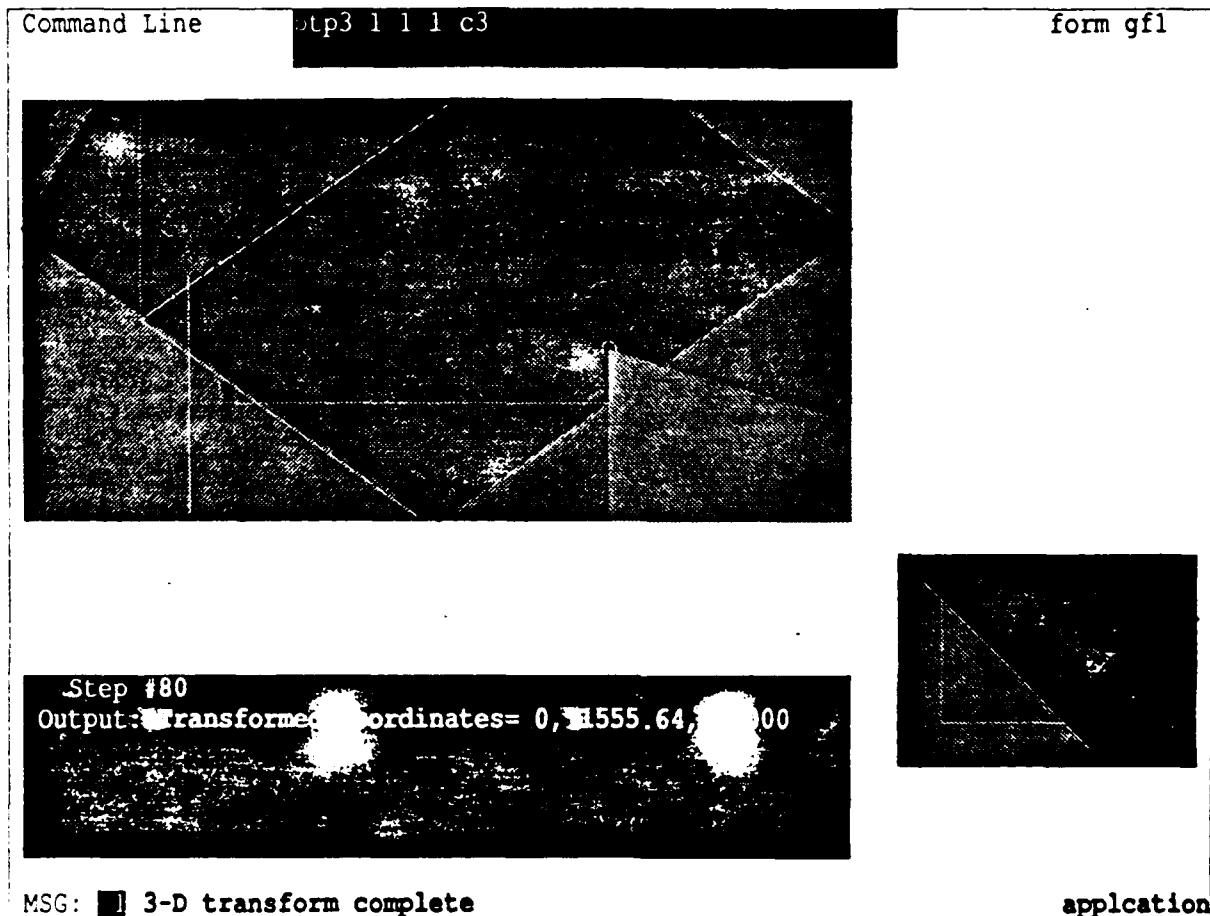
Figure 5-81b   (AFTER)

Figure 5-82a   (BEFORE)

Figure 5-82b   (AFTER)

Figure 5-83a   (BEFORE)

Figure 5-83b   (AFTER)

```
Command Line        tp3 1 1 1 b3                              form gf1
```
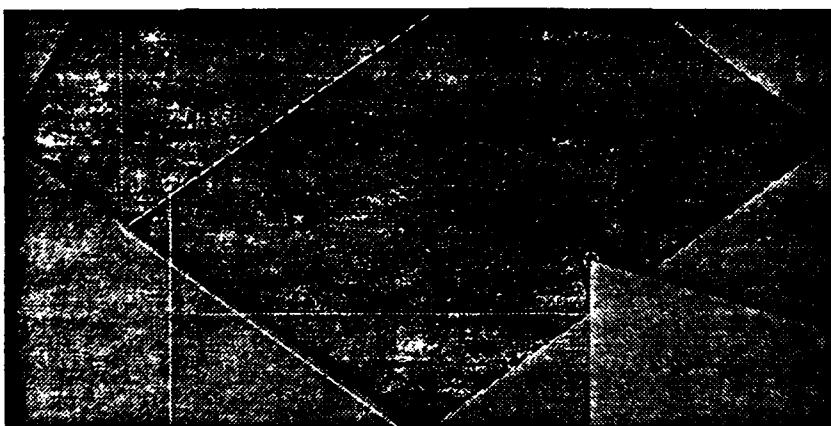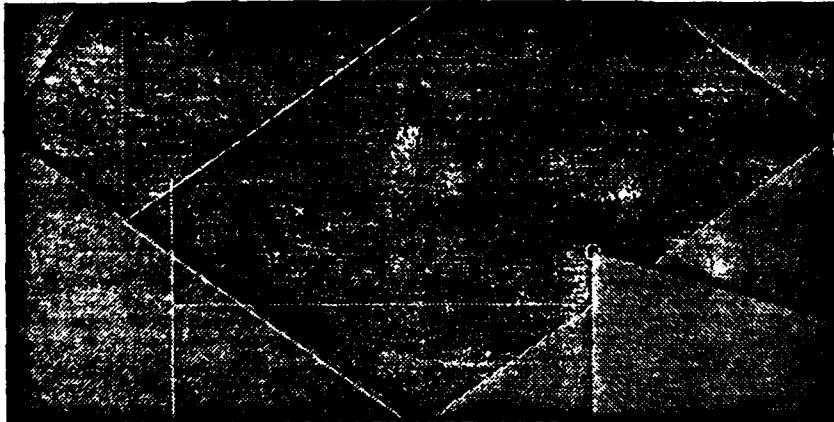
Step #82
Output: Transformed coordinates= 0, 1555.64, 11000

MSG: ▮ 3-D compose transformation matrix complete   applcation
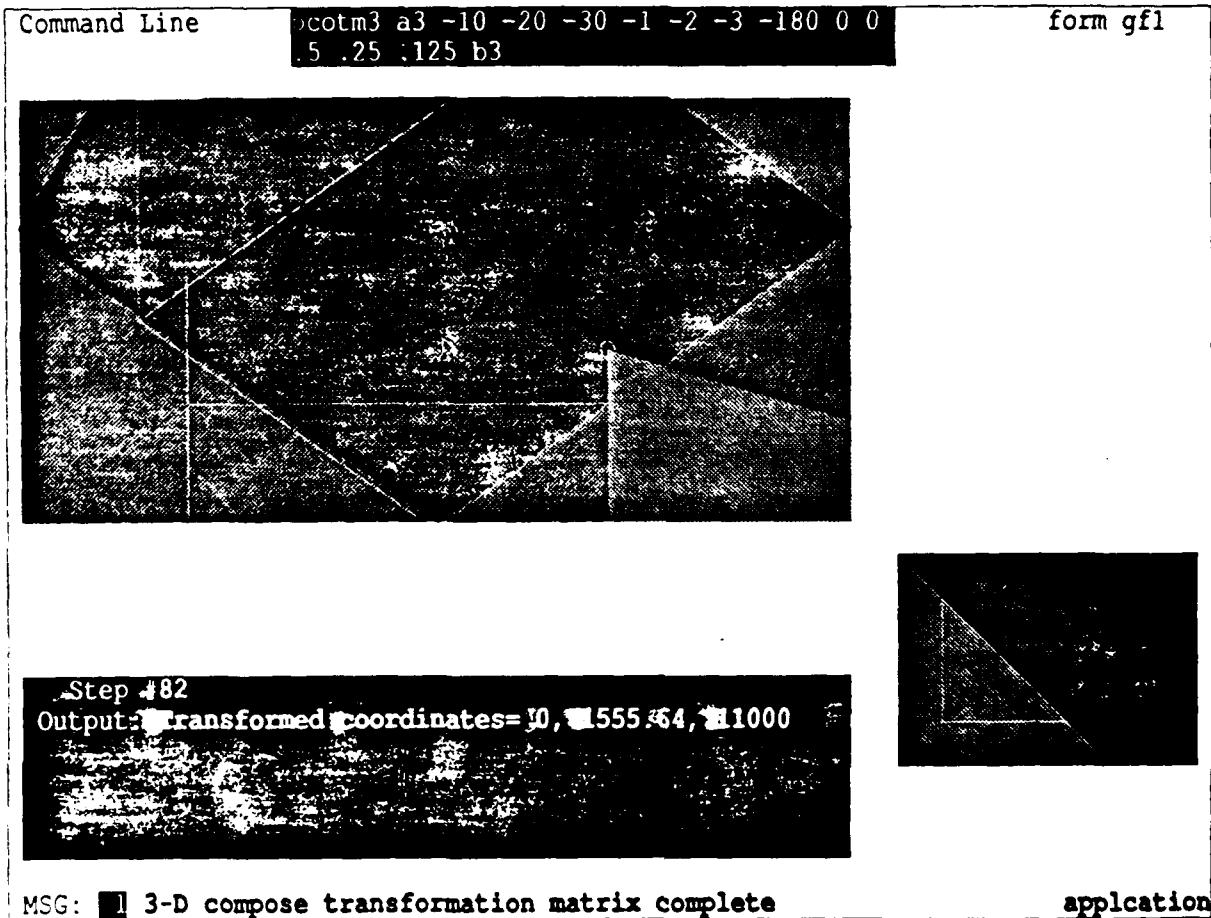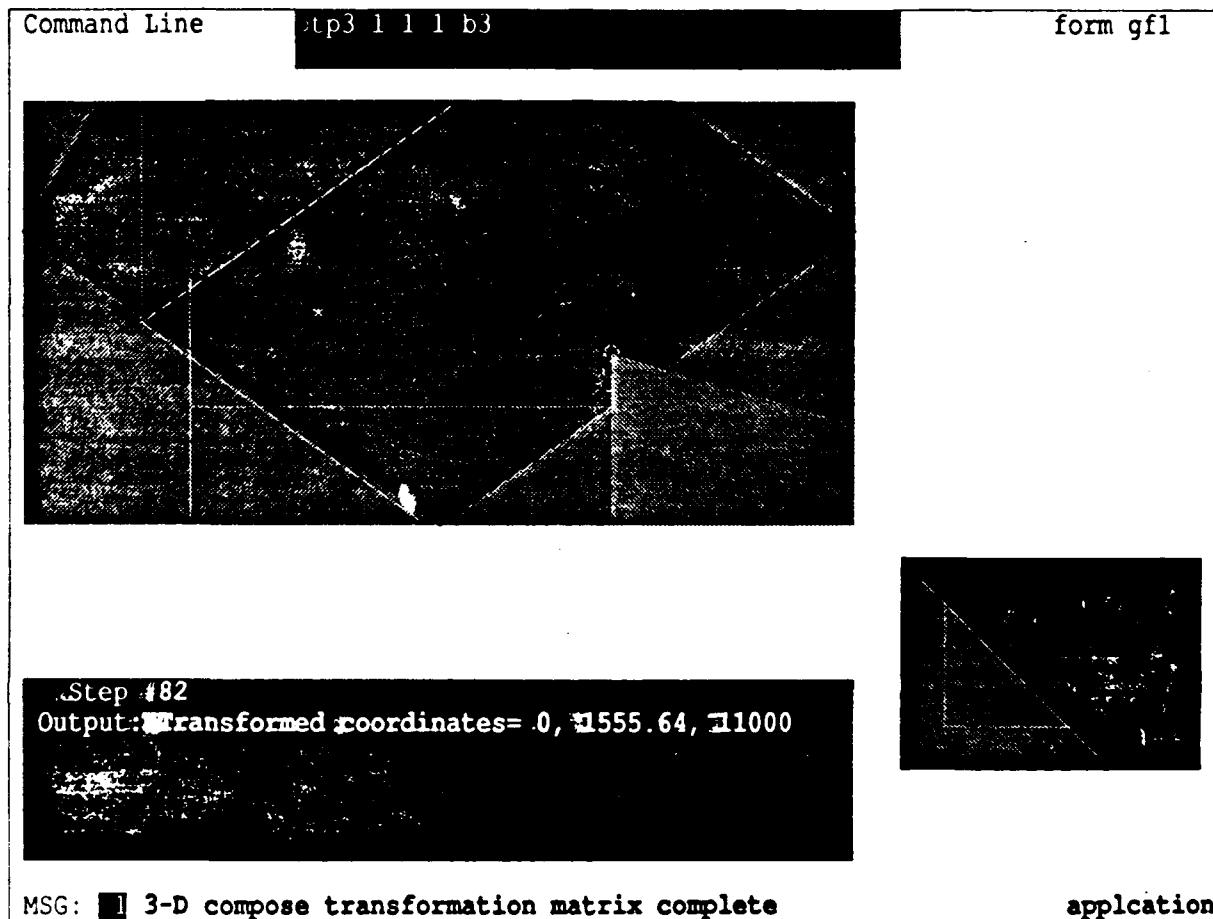
Figure 5-84a   (BEFORE)

Figure 5-84b   (AFTER)

```
Command Line        swkw3 1 0 1 0 .69999237 0 1                form gfl
```
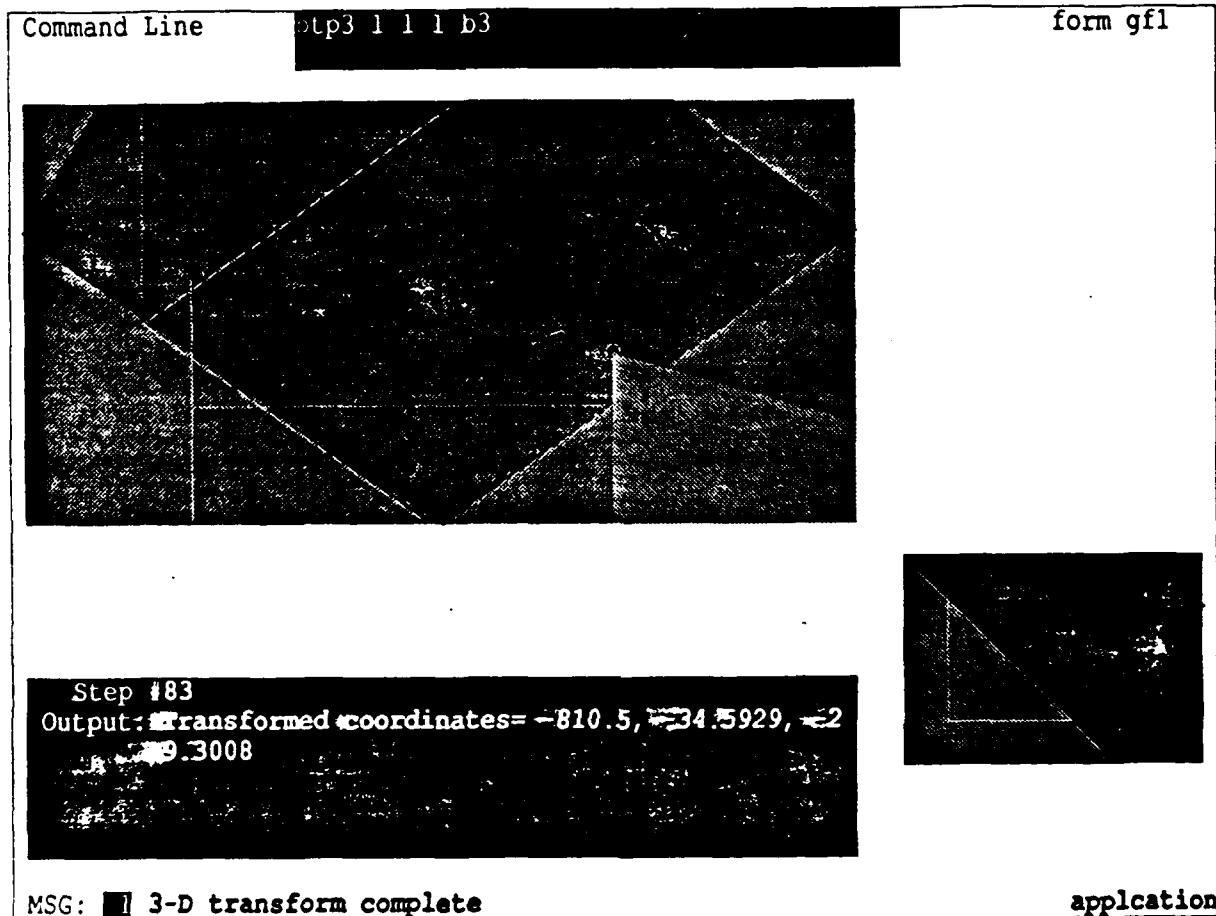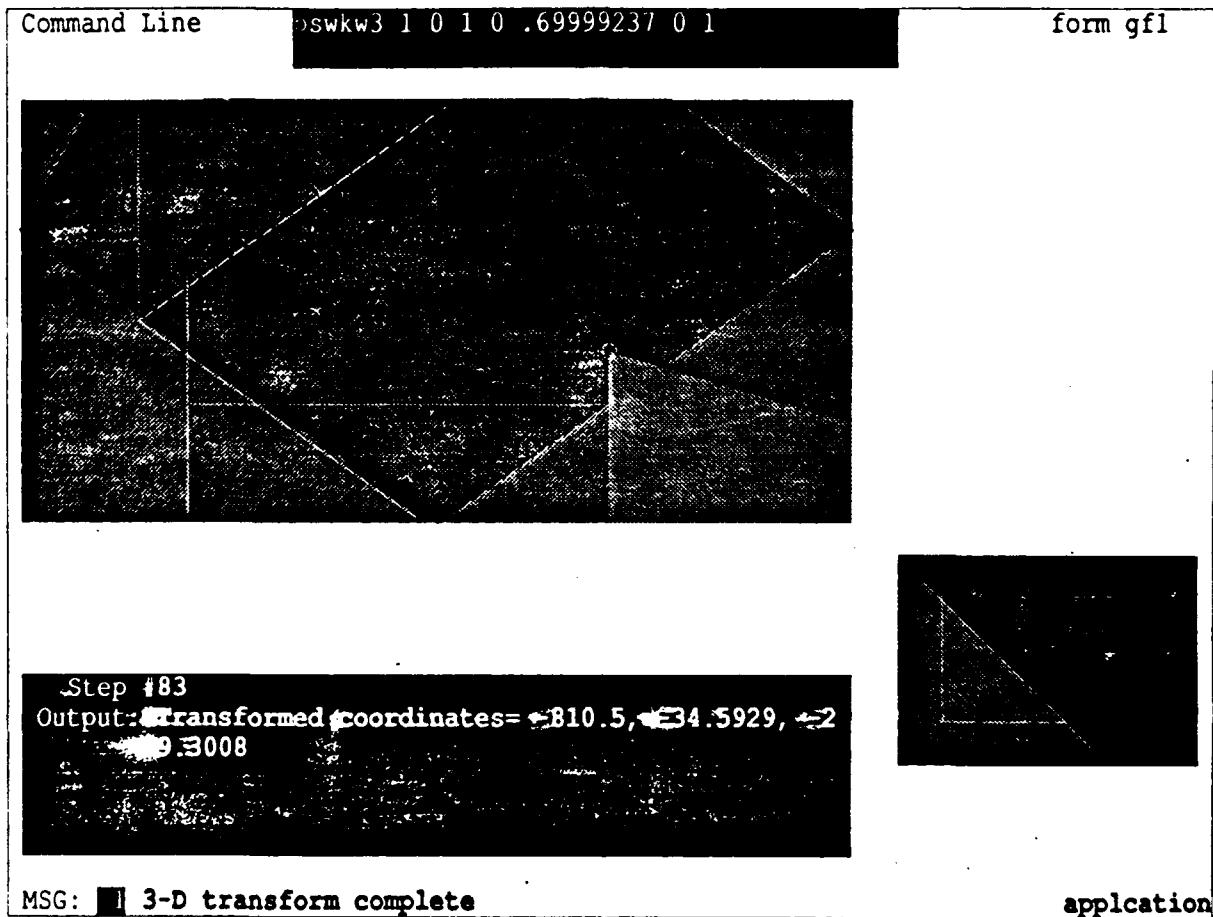
Figure 5-85a   (BEFORE)
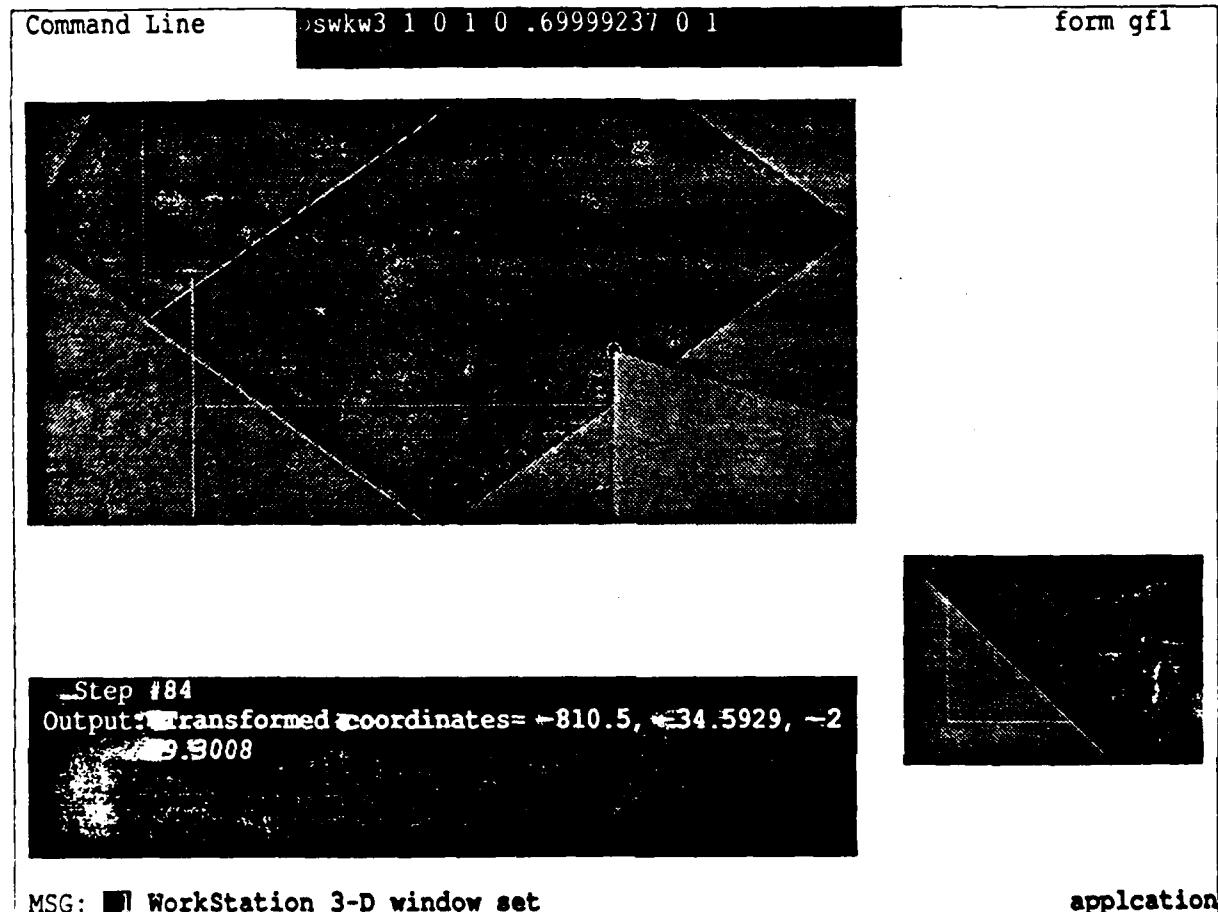
Figure 5-85b   (AFTER)

Figure 5-86a   (BEFORE)

```
Command Line       swkv3 1 0 65535 0 33363 0 65535              form gfl
```
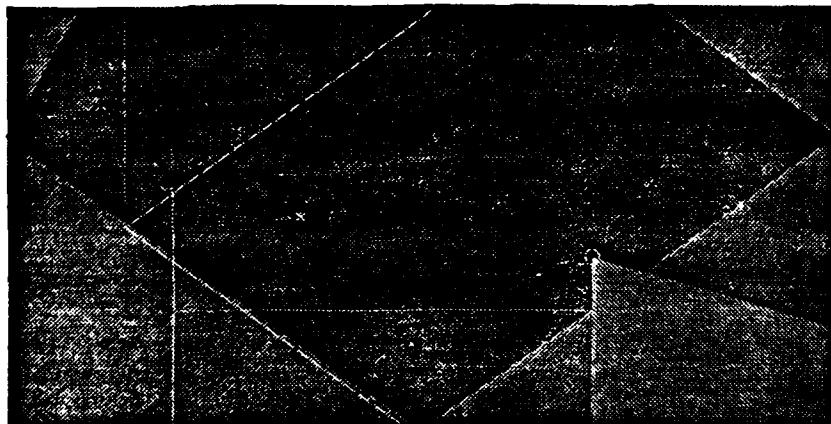
Step #85
Output: Transformed coordinates= -810.5, -34.5929, -2
9.3008

MSG: ▮] WorkStation 3-D viewport set   applcation
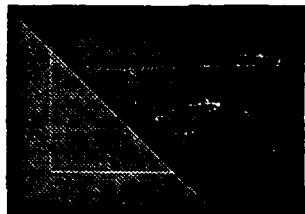
Figure 5-86b   (AFTER)
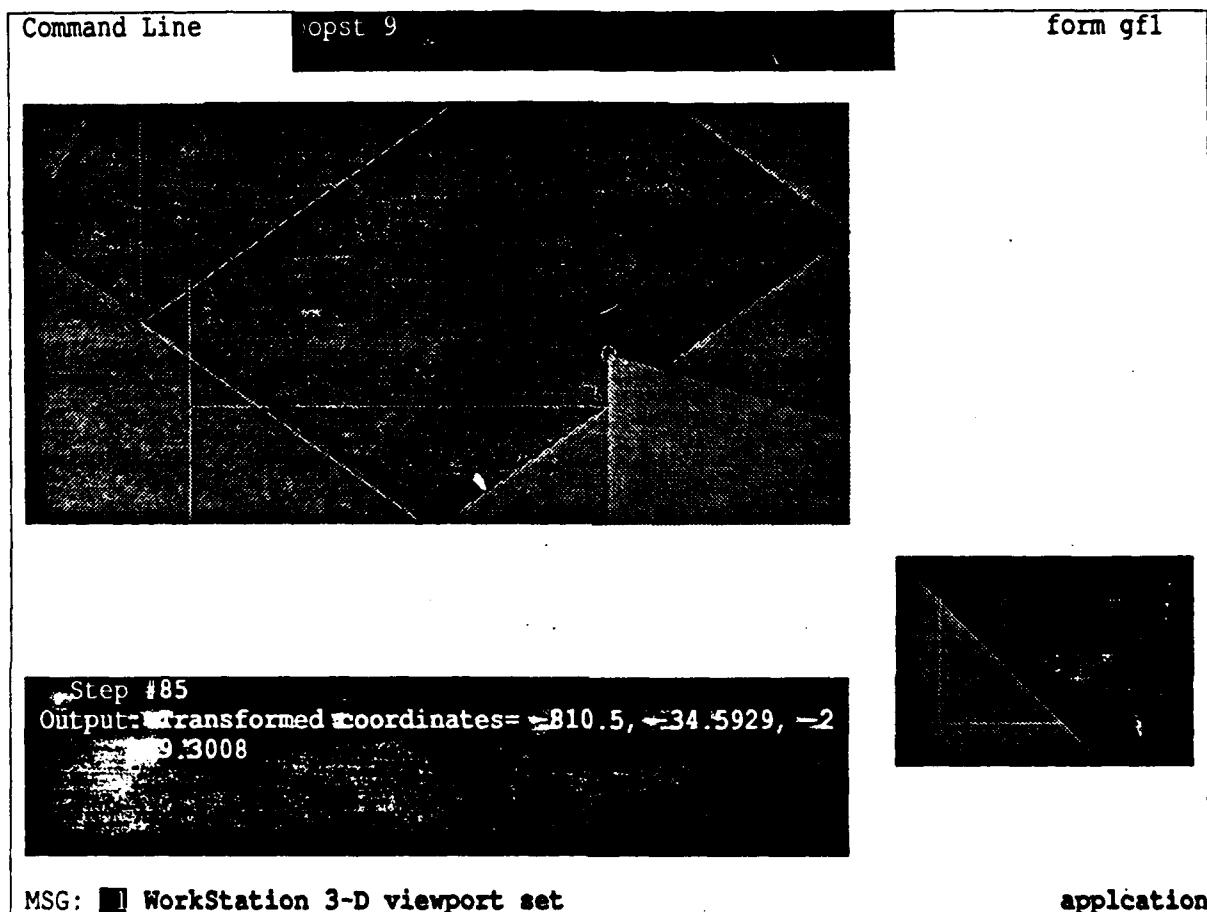
Figure 5-87a   (BEFORE)
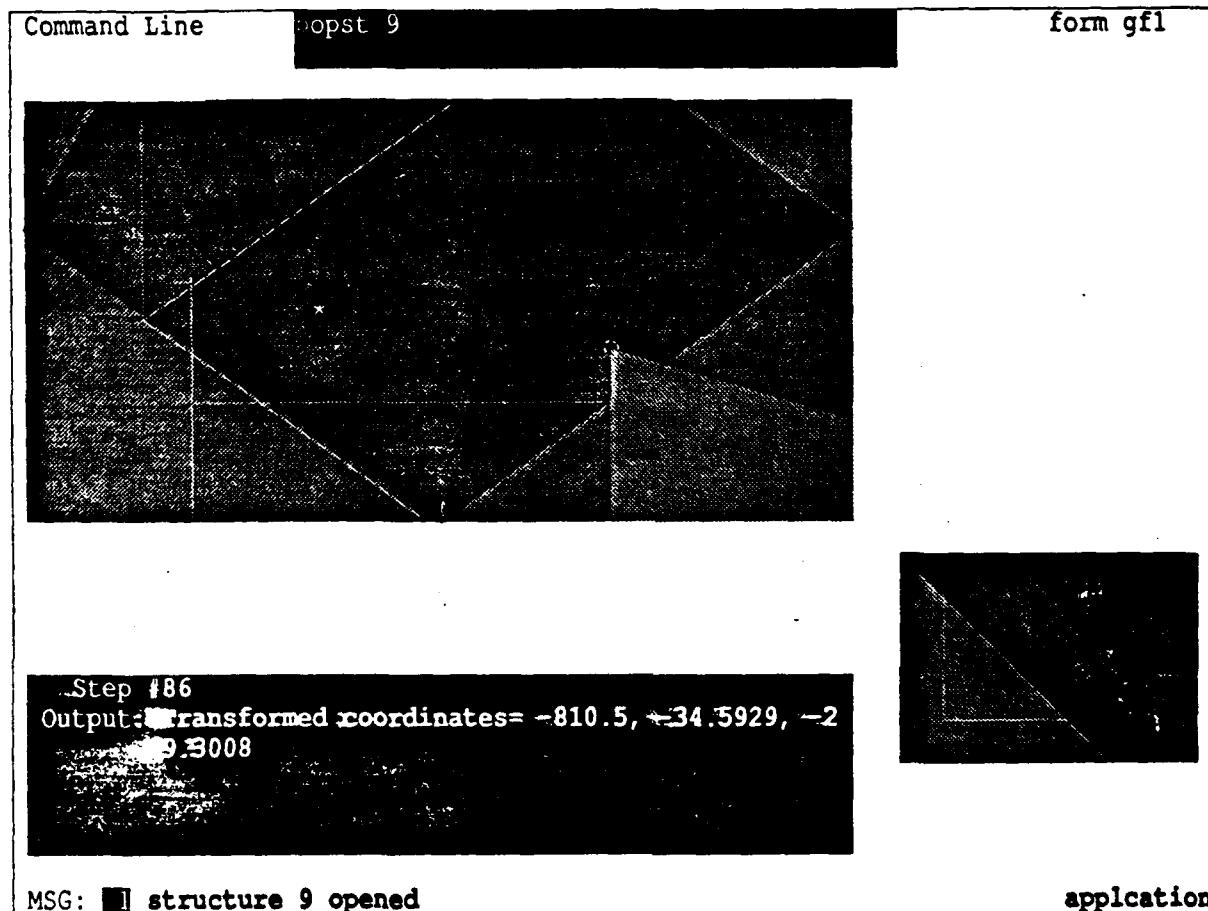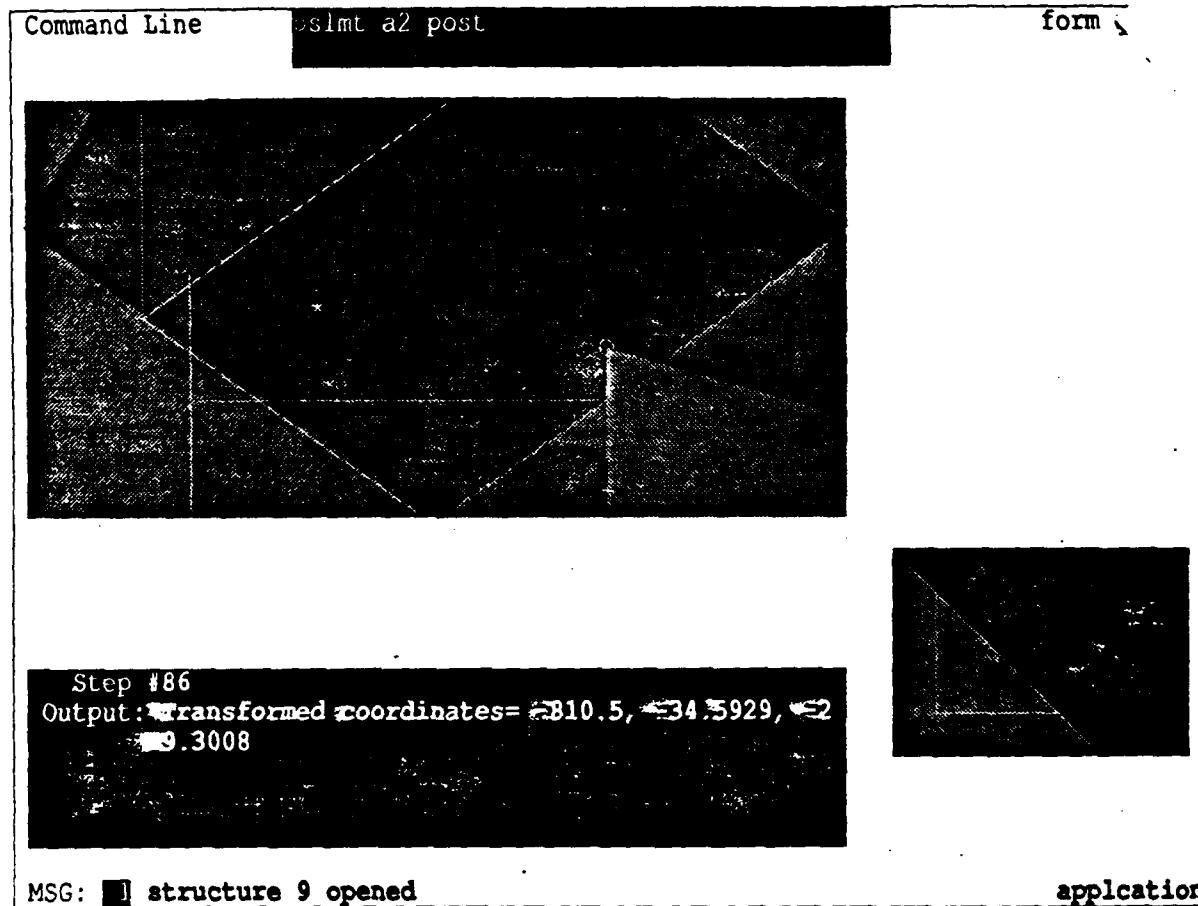
Figure 5-87b   (AFTER)

Figure 5-88a   (BEFORE)
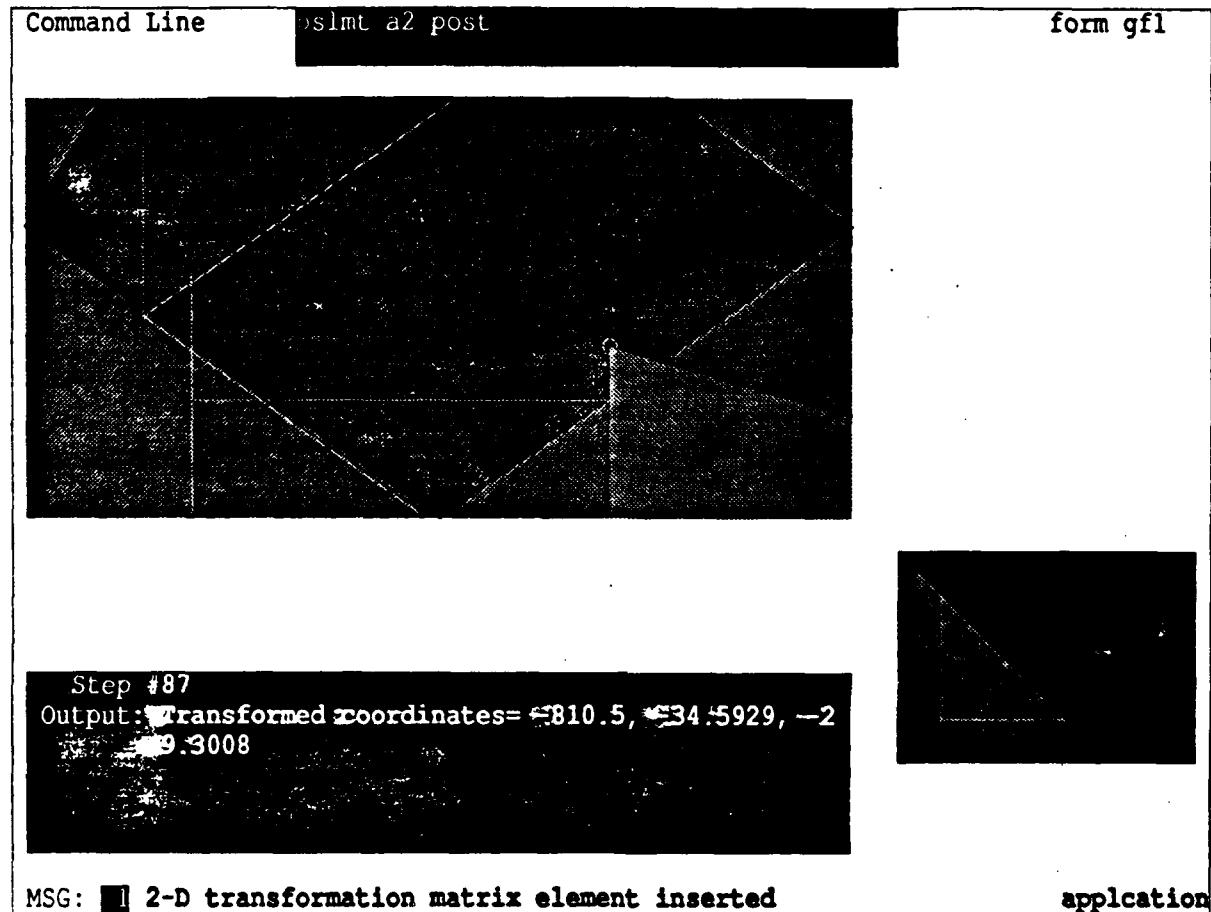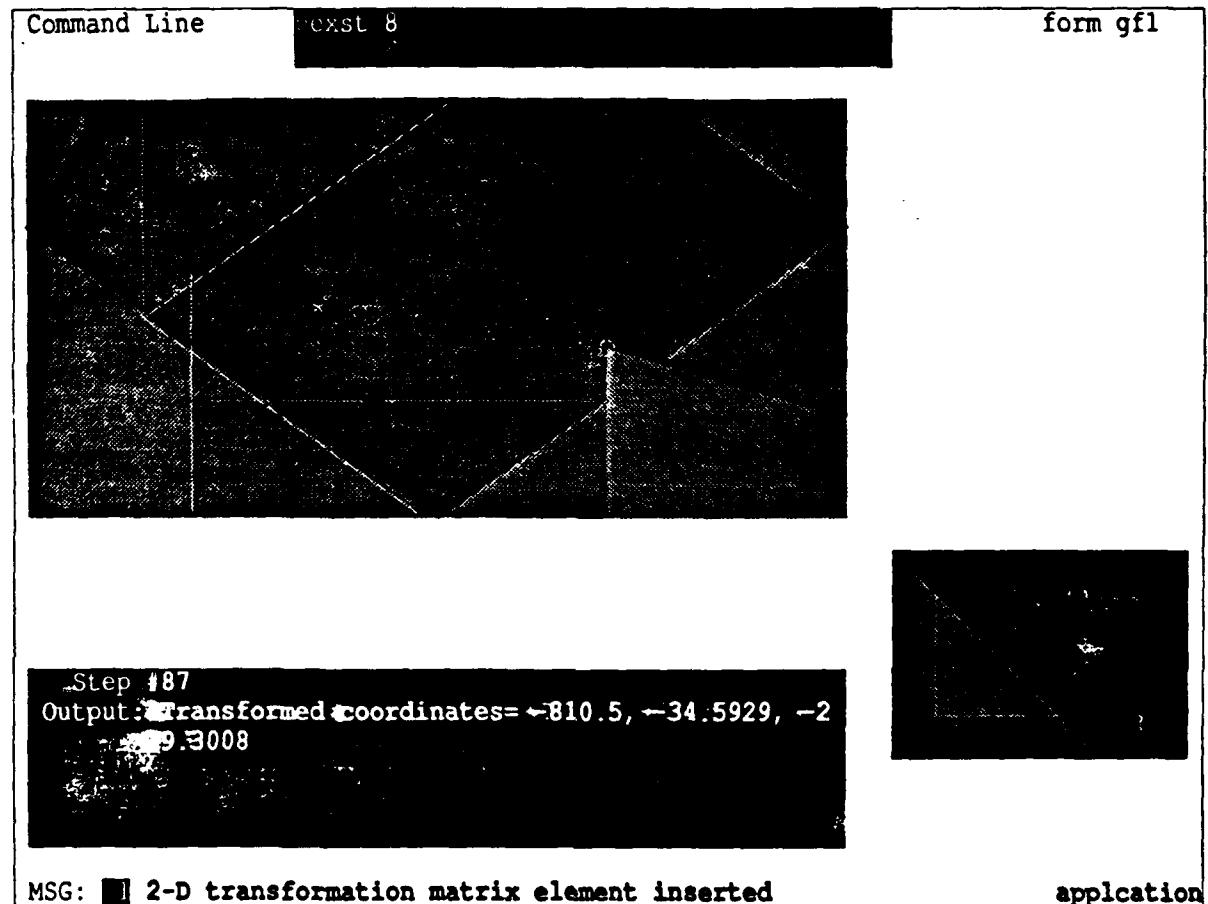
Figure 5-88b (AFTER)
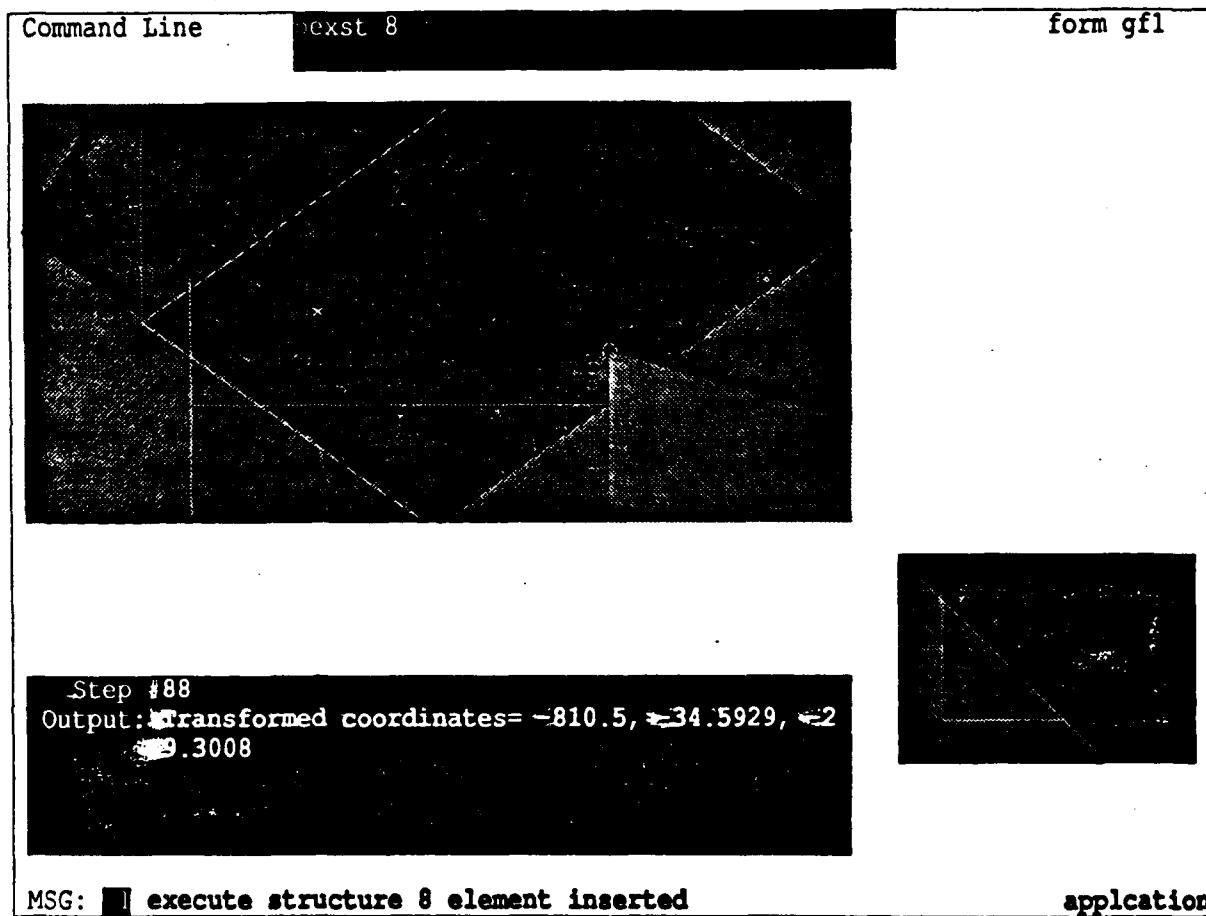
Figure 5-89a   (BEFORE)

Figure 5-89b   (AFTER)
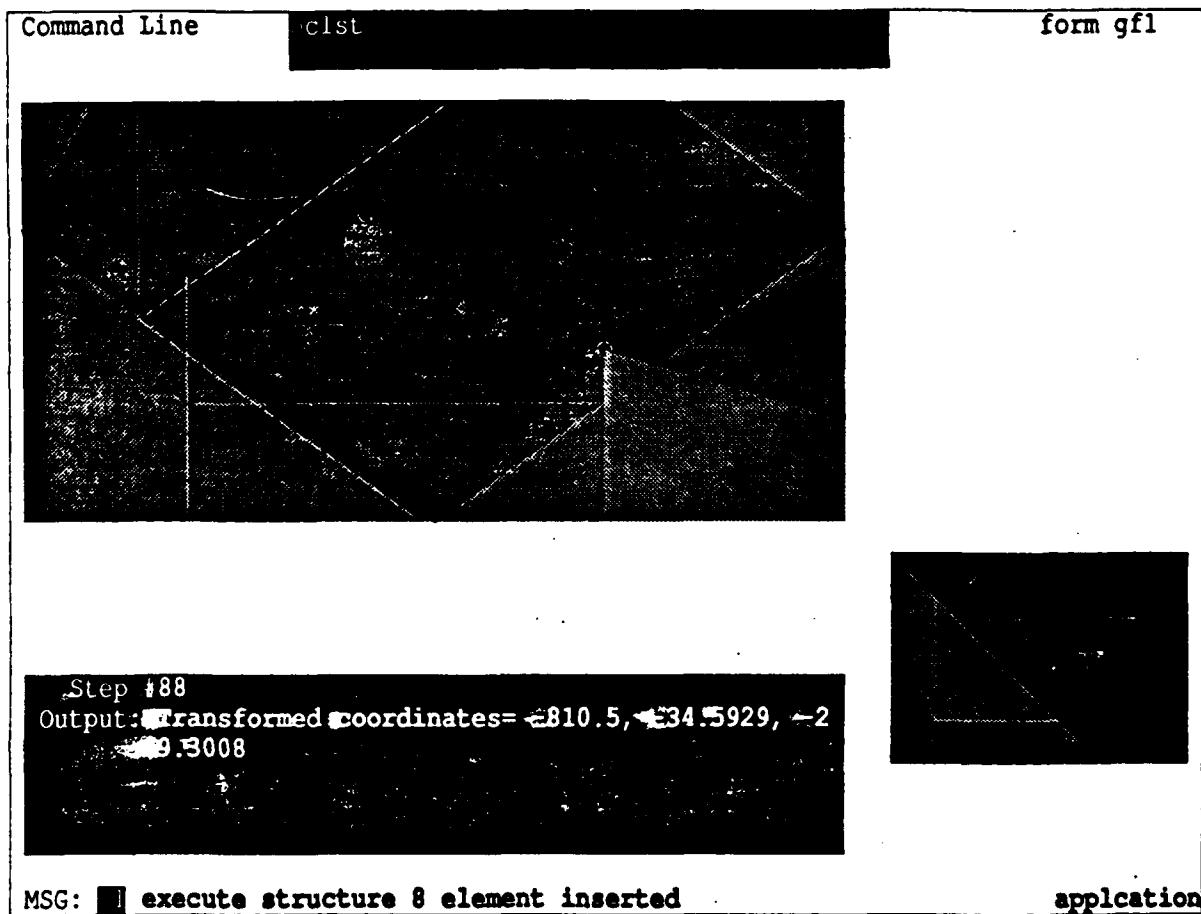
Figure 5-90a   (BEFORE)
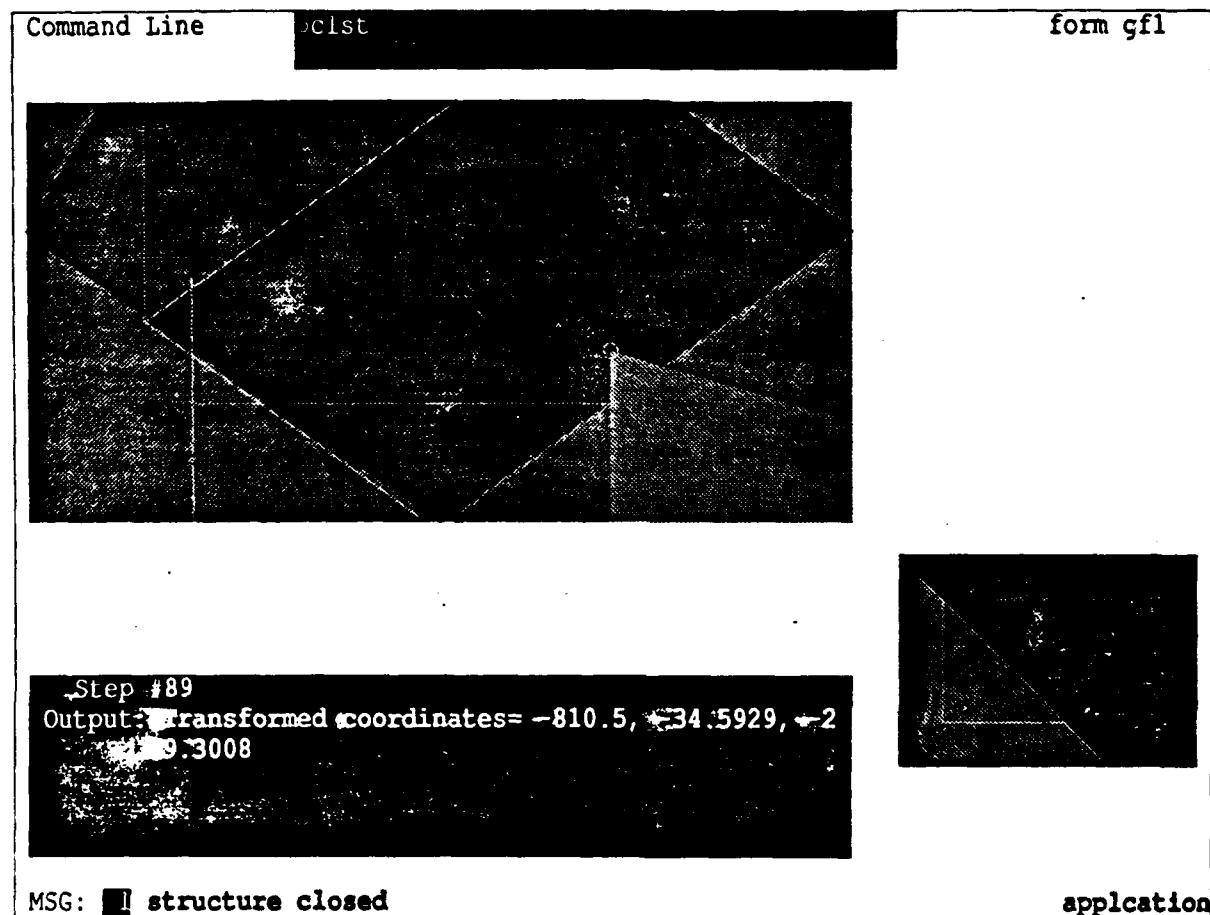
Figure 5-90b   (AFTER)
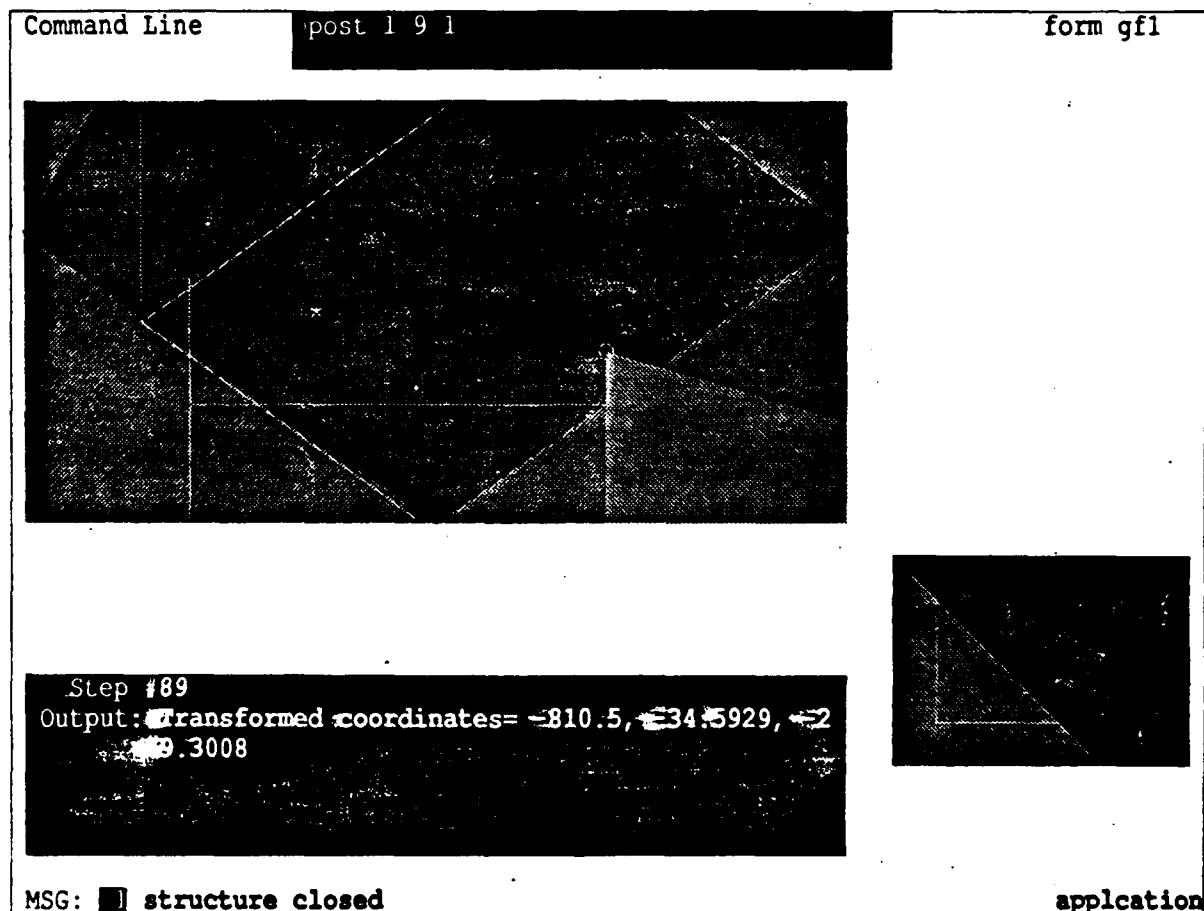
Figure 5-91a   (BEFORE)
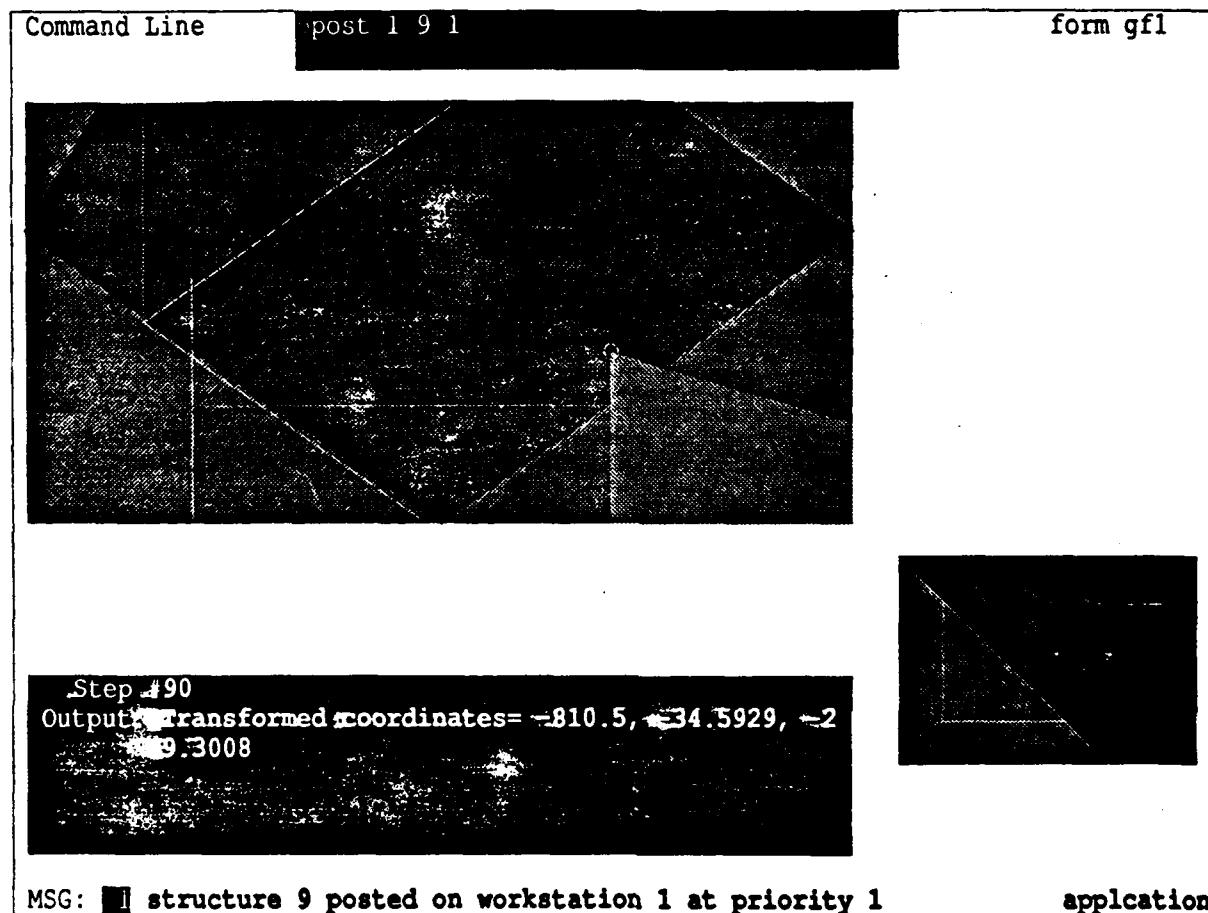
Figure 5-91b   (AFTER)
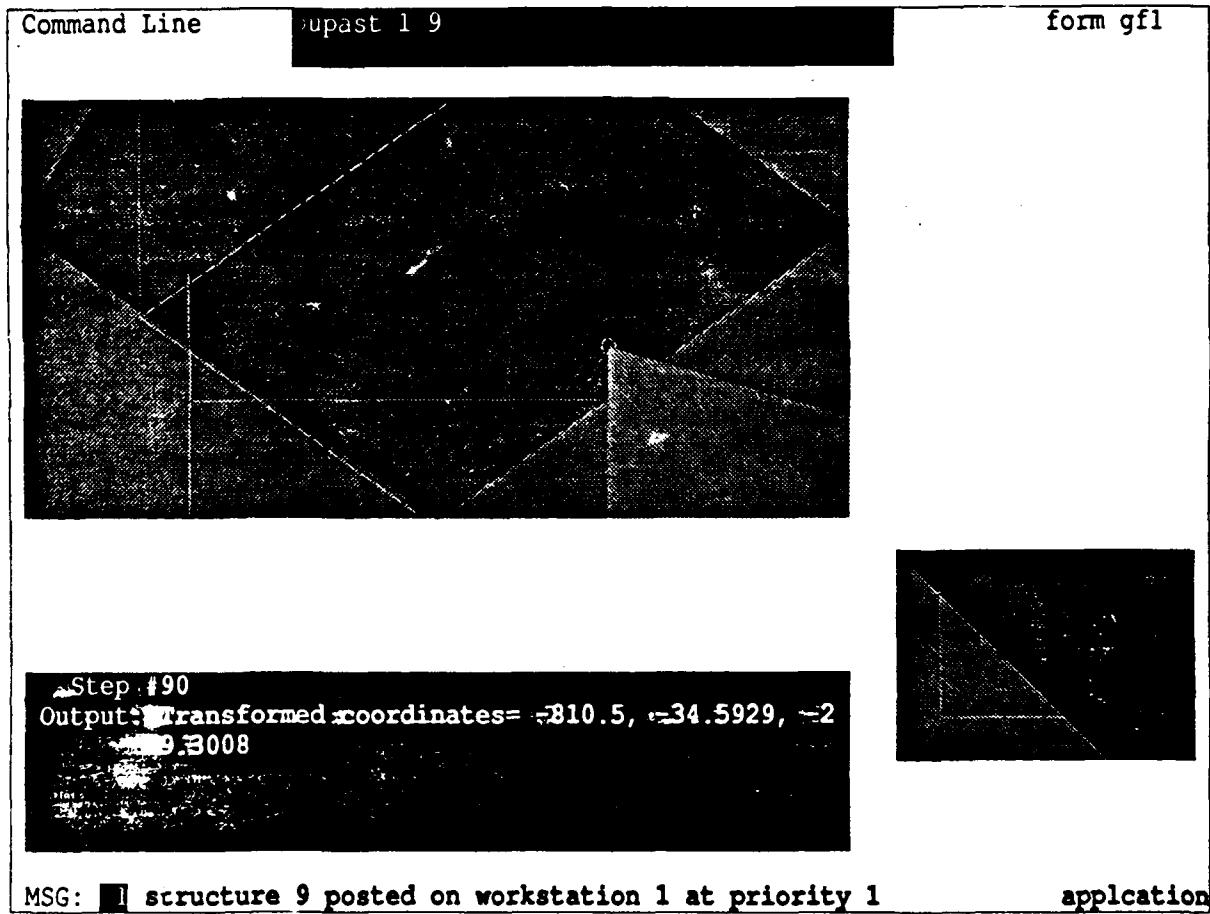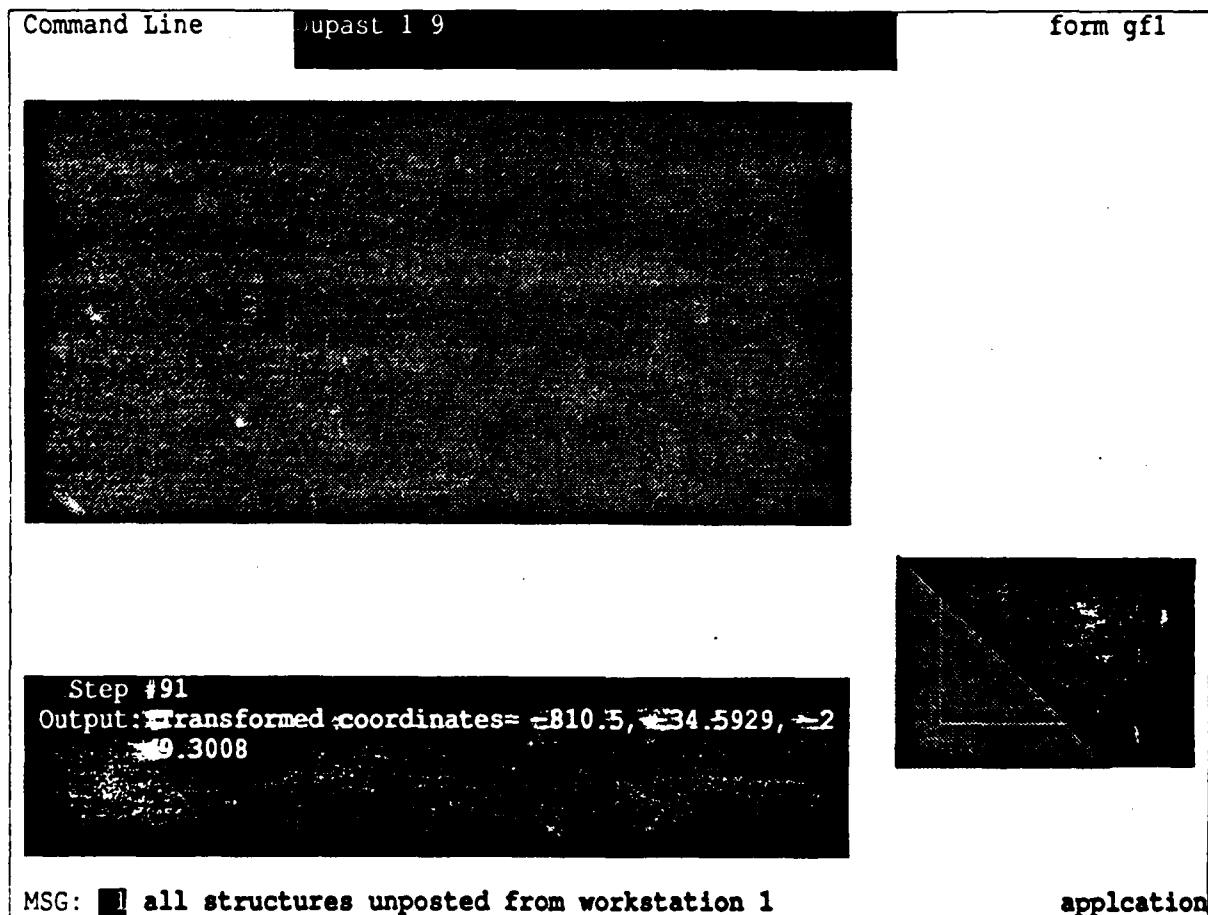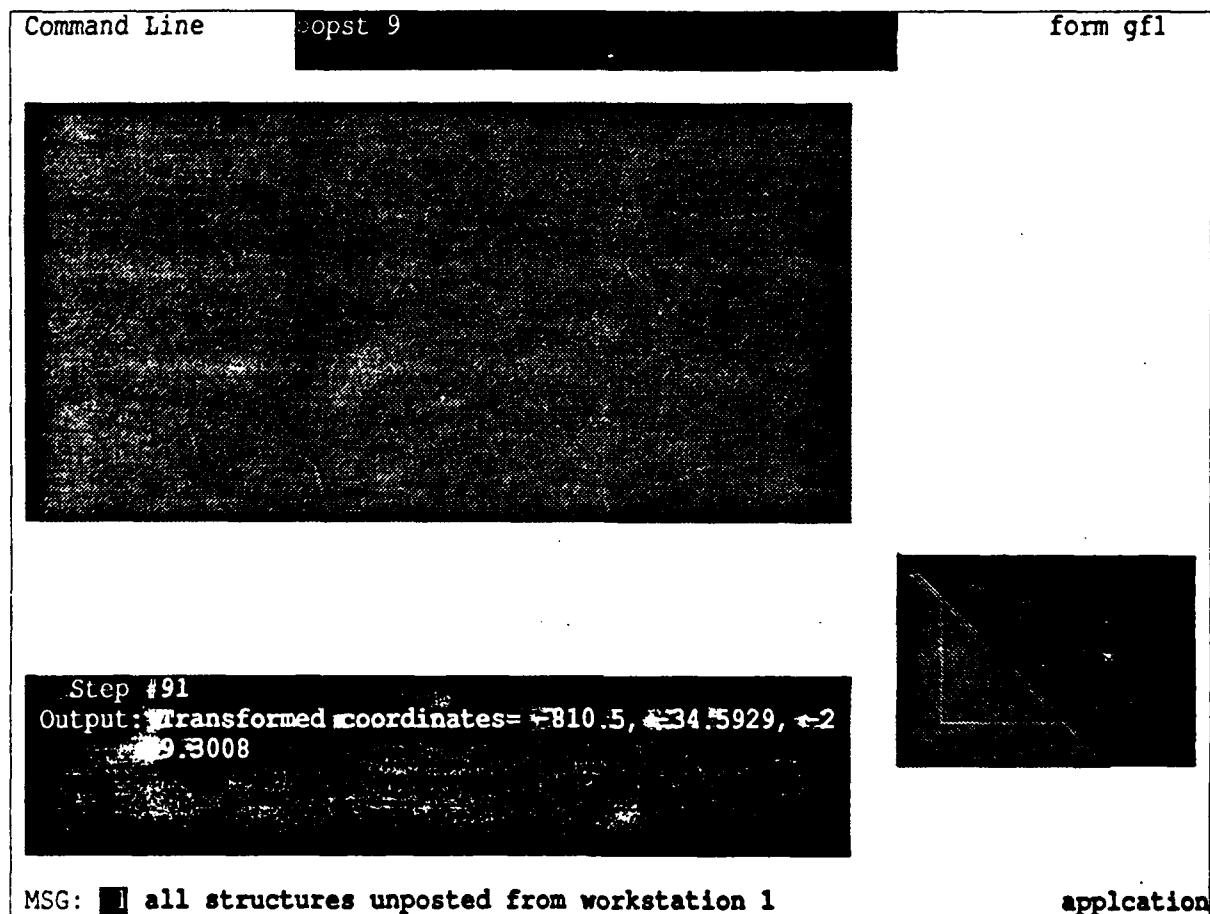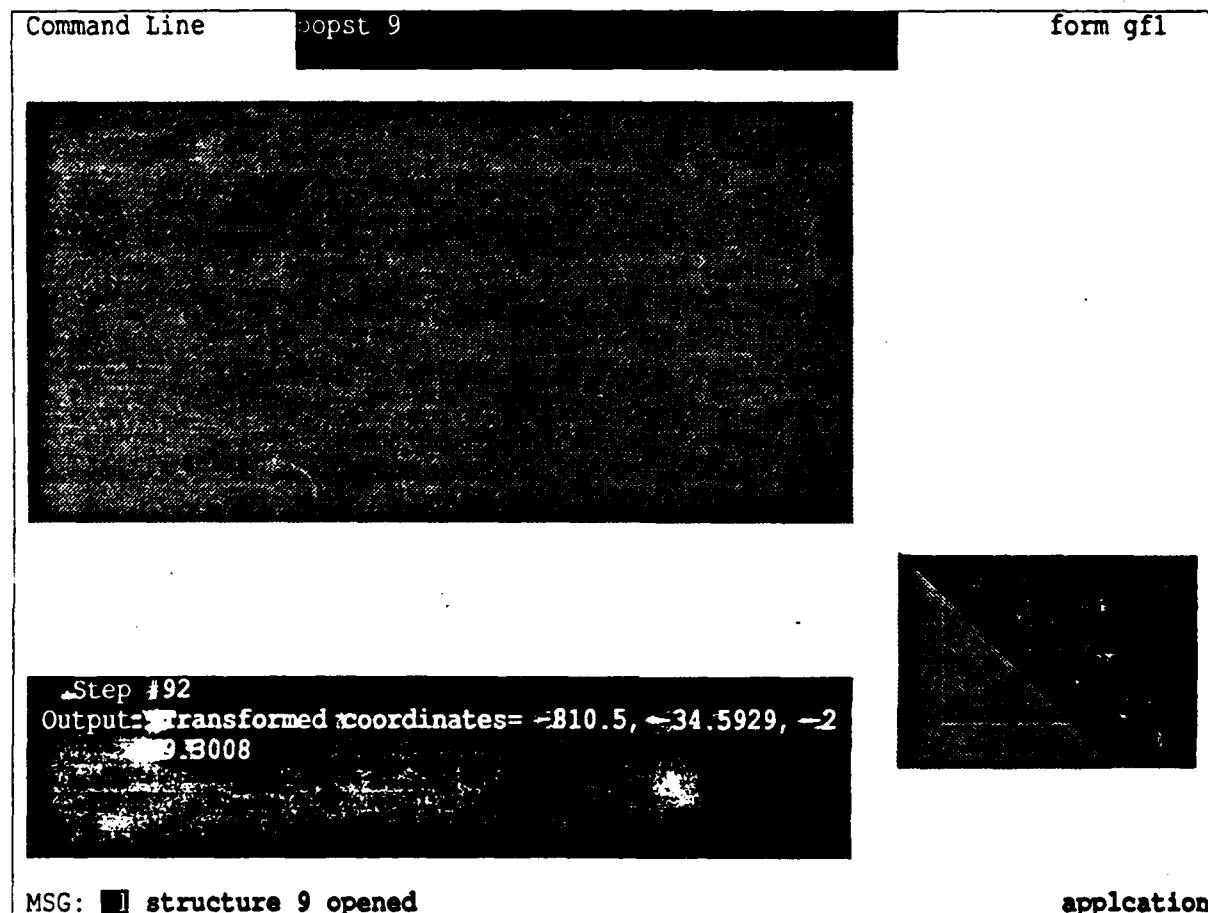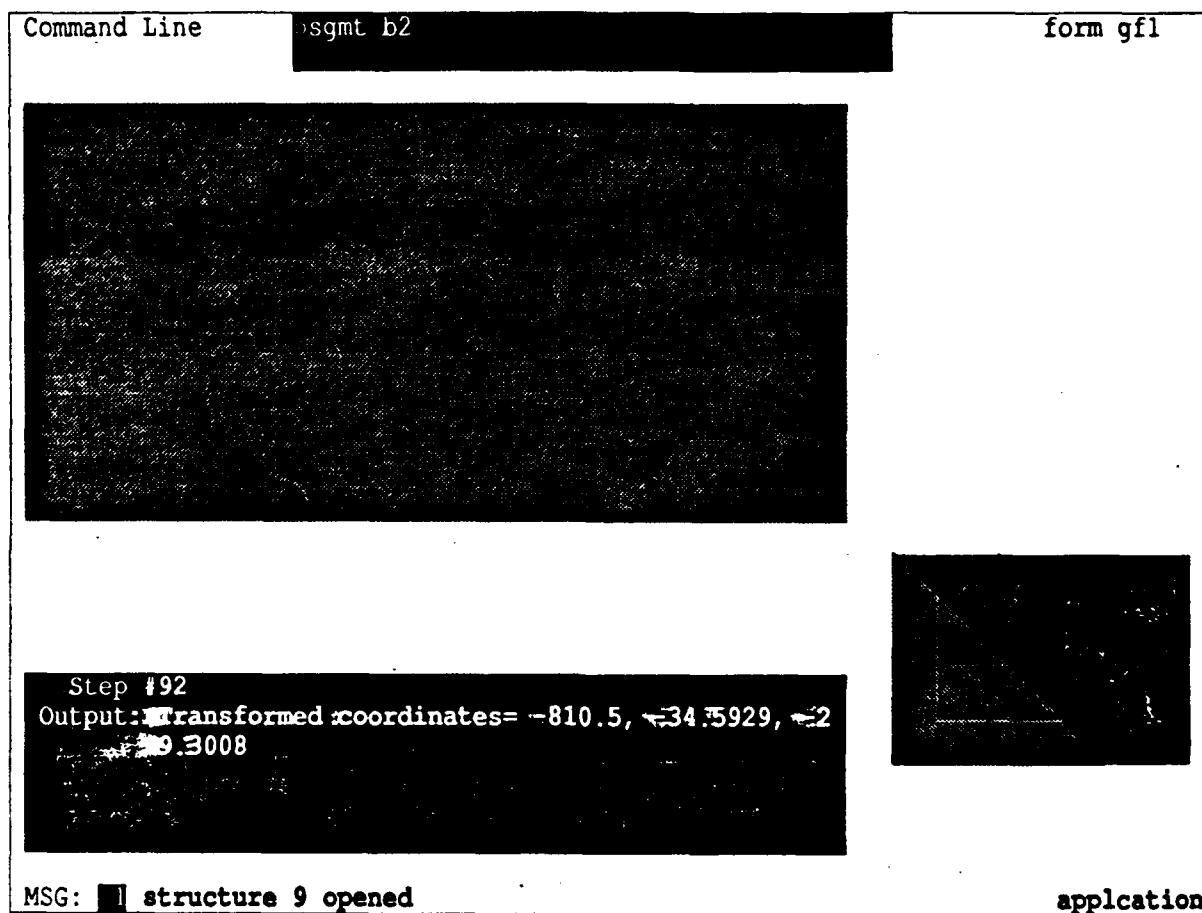
Figure 5-92a   (BEFORE)

Figure 5-92b   (AFTER)

Figure 5-93a   (BEFORE)
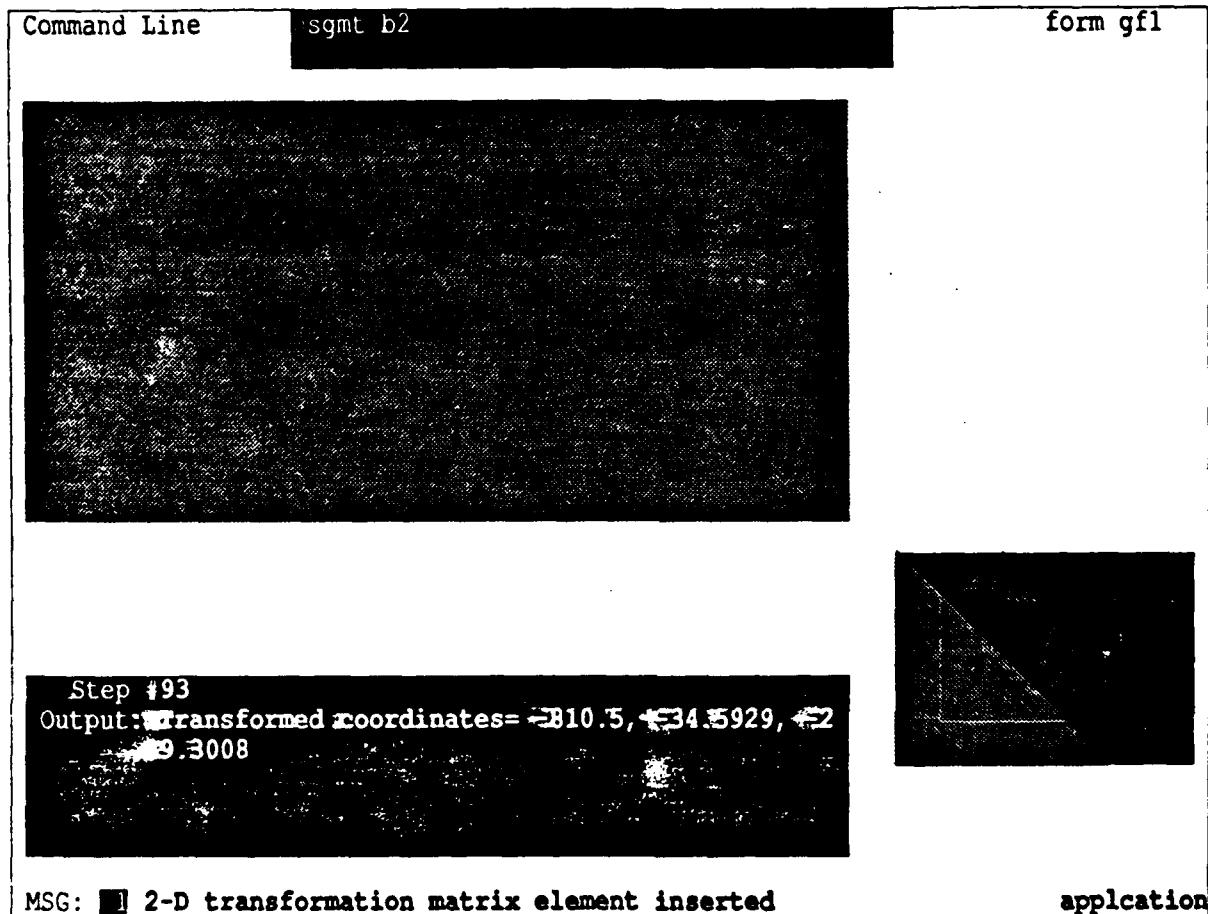
Figure 5-93b   (AFTER)

Figure 5-94a   (BEFORE)
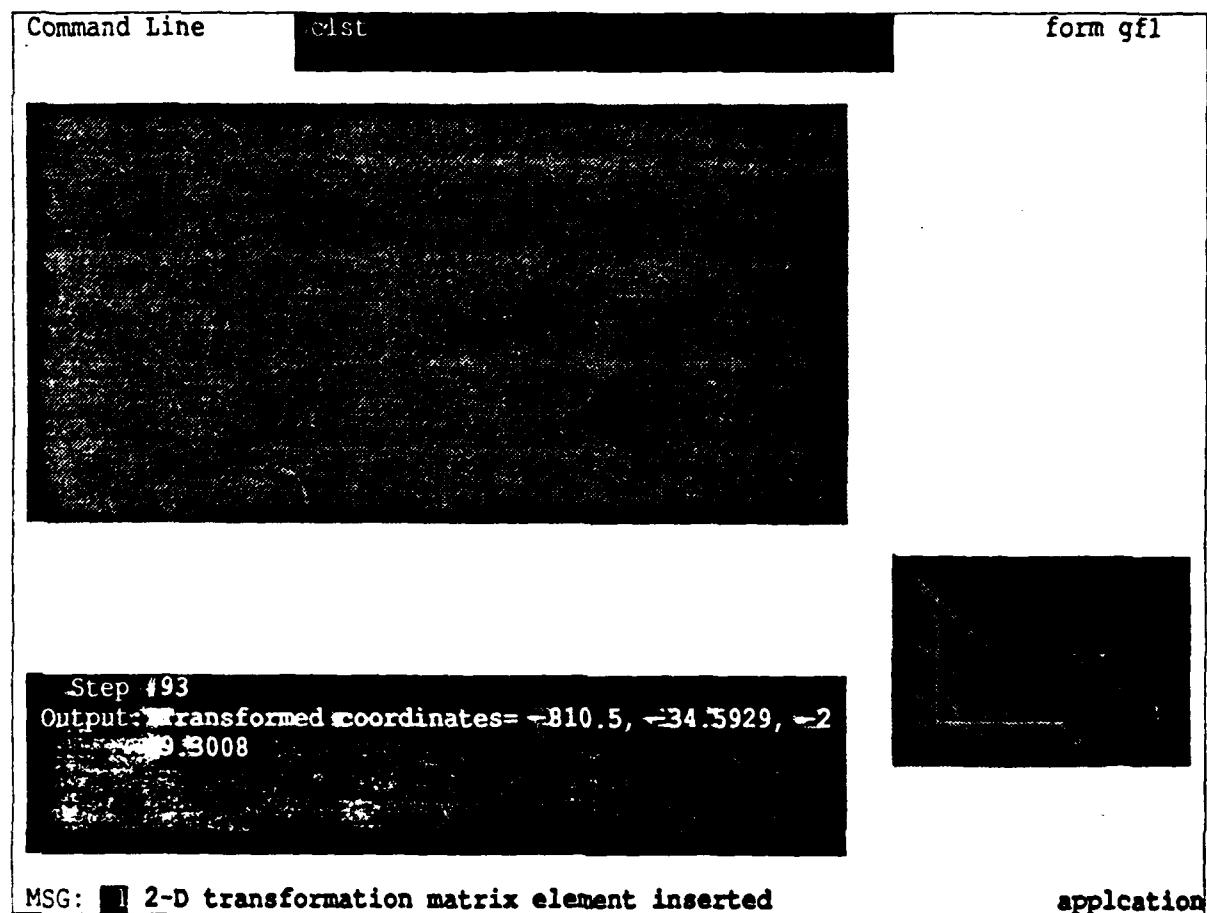
Figure 5-94b   (AFTER)
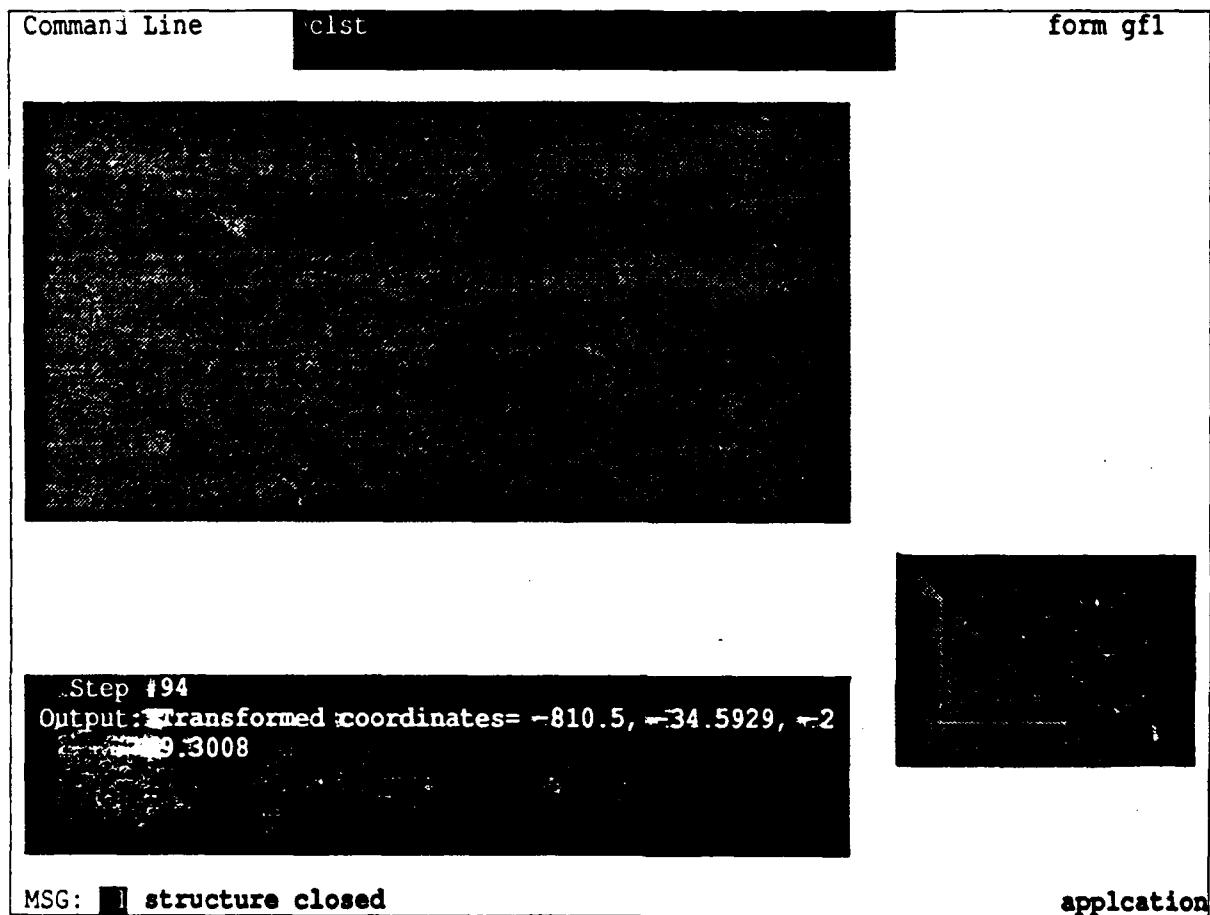
Figure 5-95a   (BEFORE)
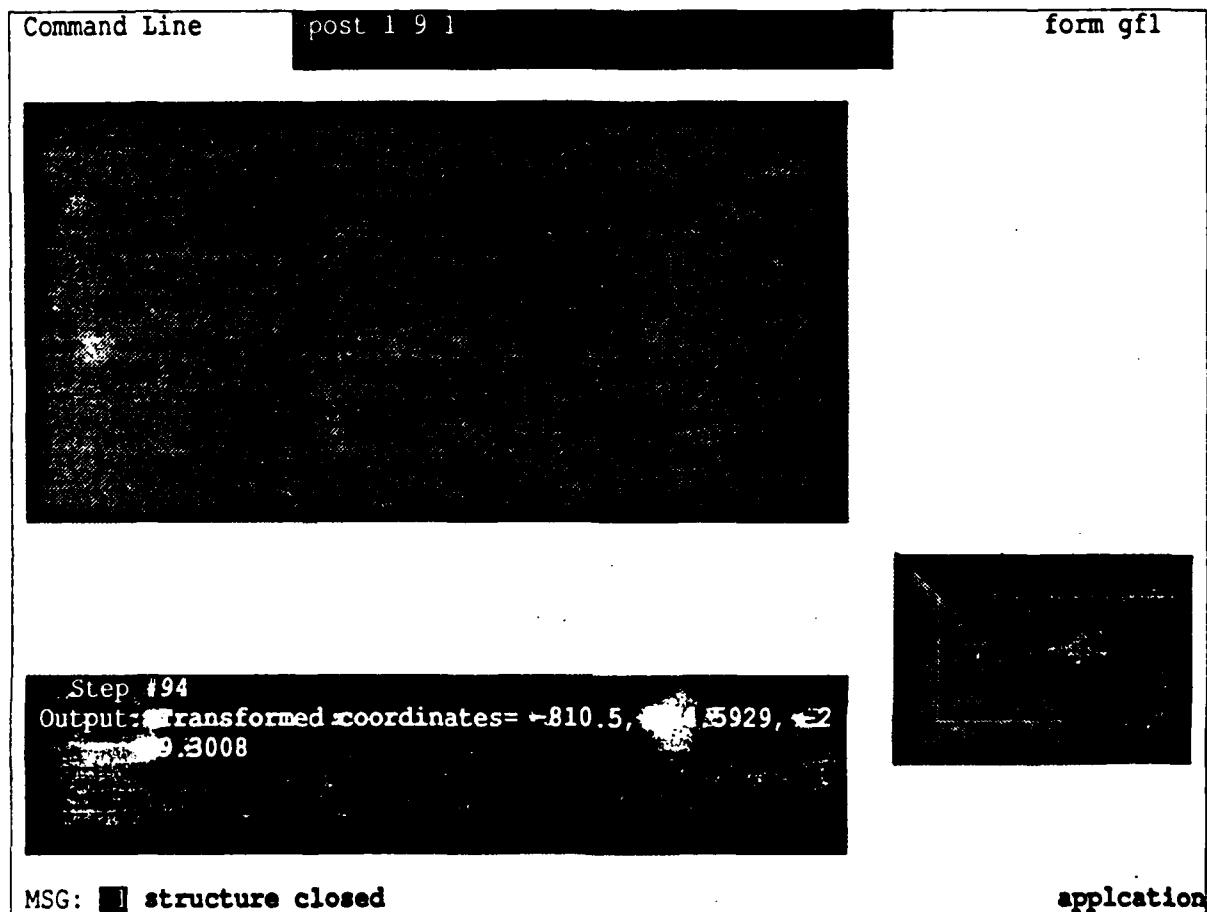
Figure 5-95b   (AFTER)

Figure 5-96a   (BEFORE)
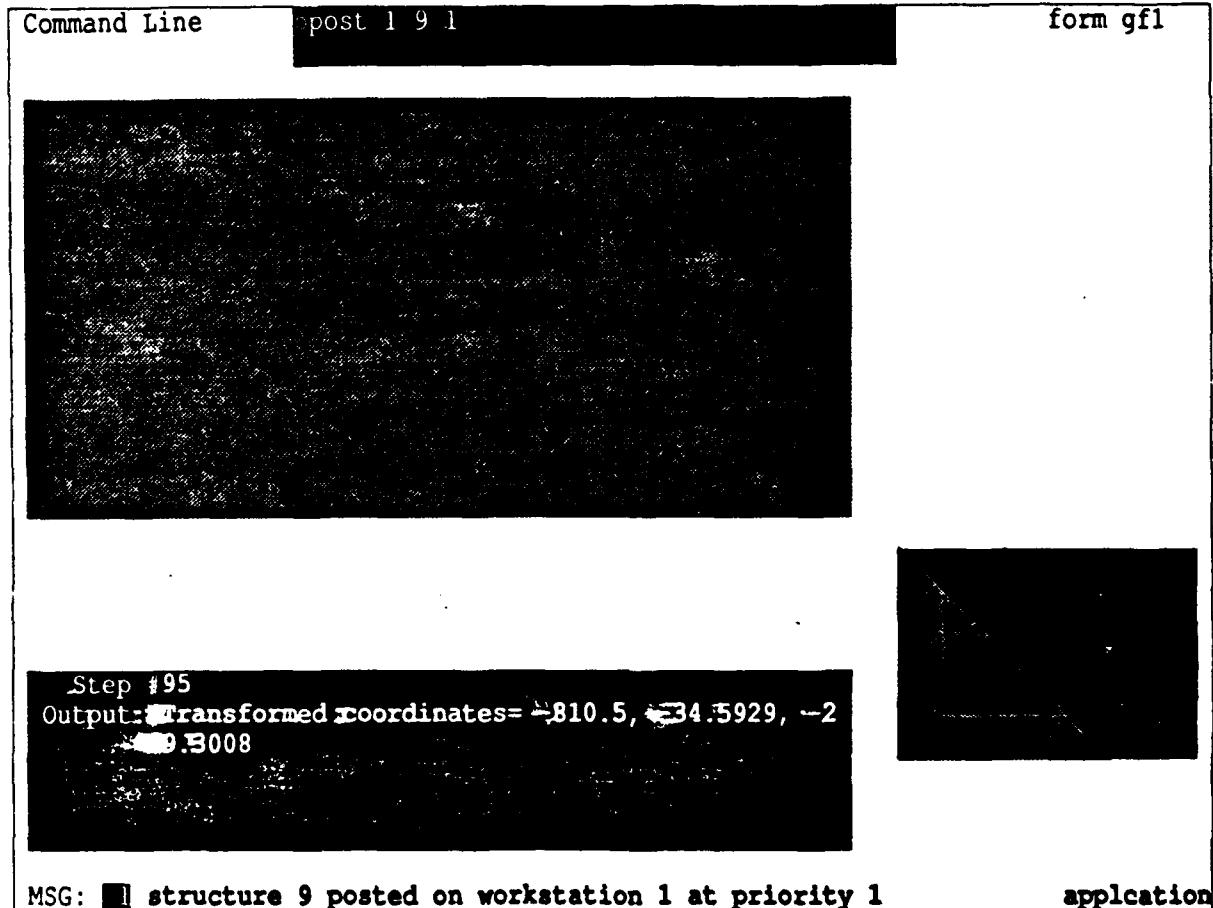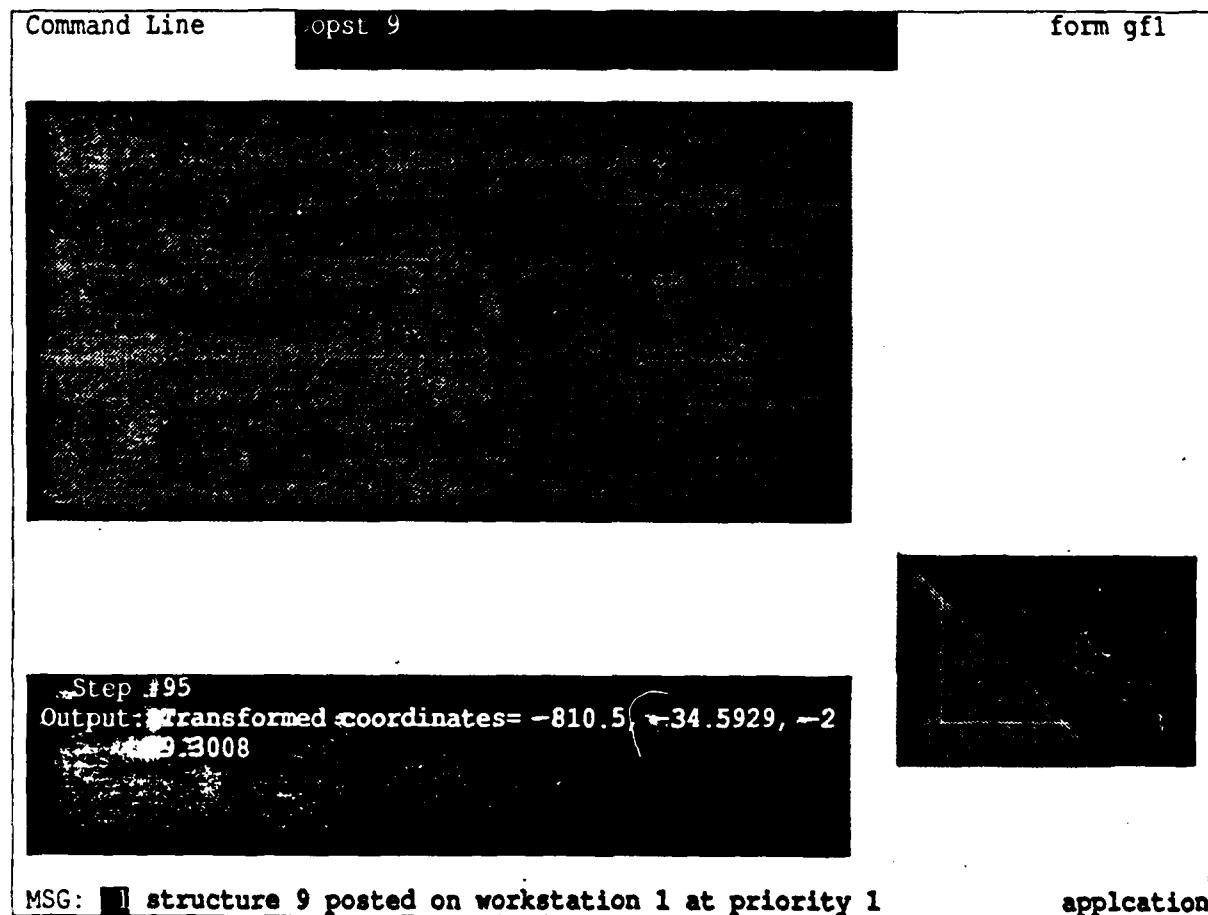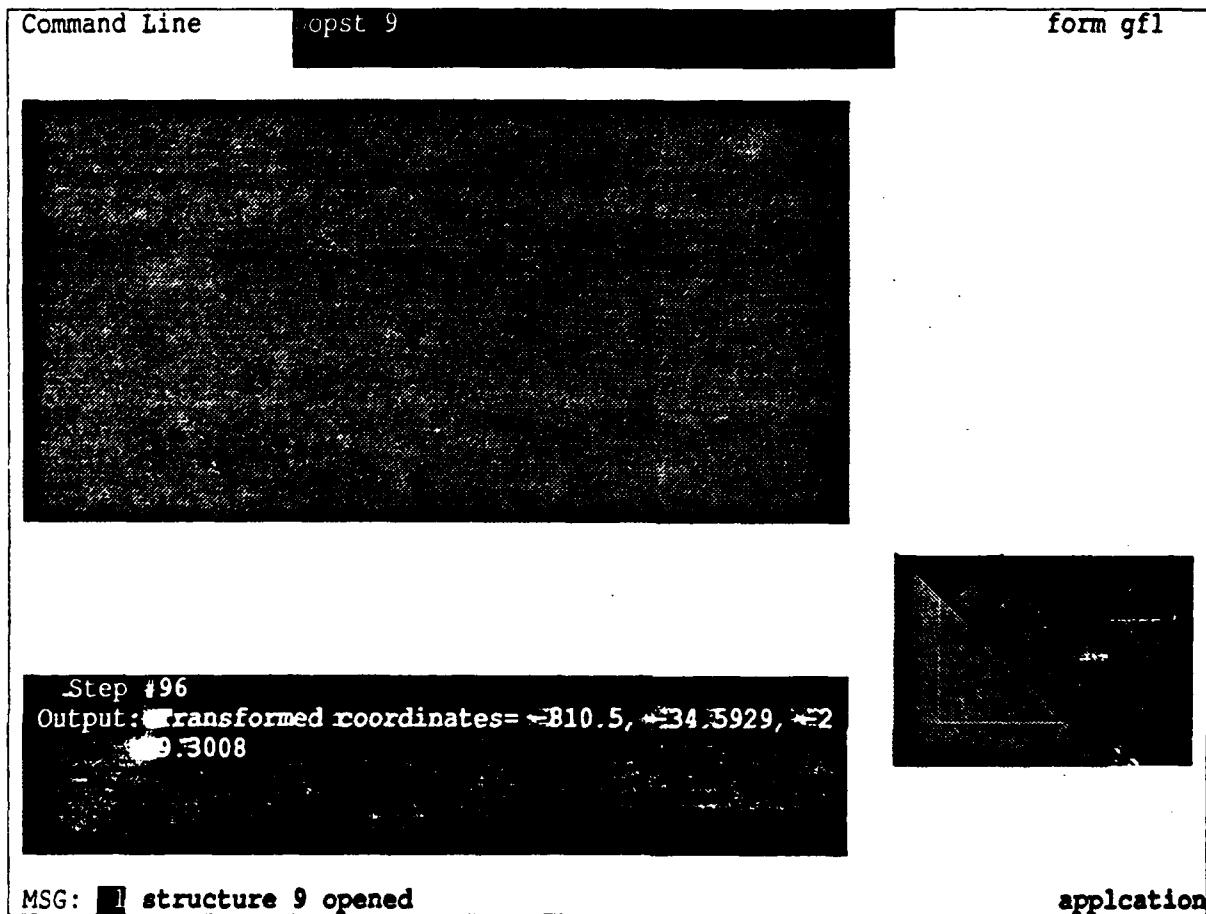
Figure 5-96b   (AFTER)

Figure 5-97a (BEFORE)

Figure 5-97b   (AFTER)
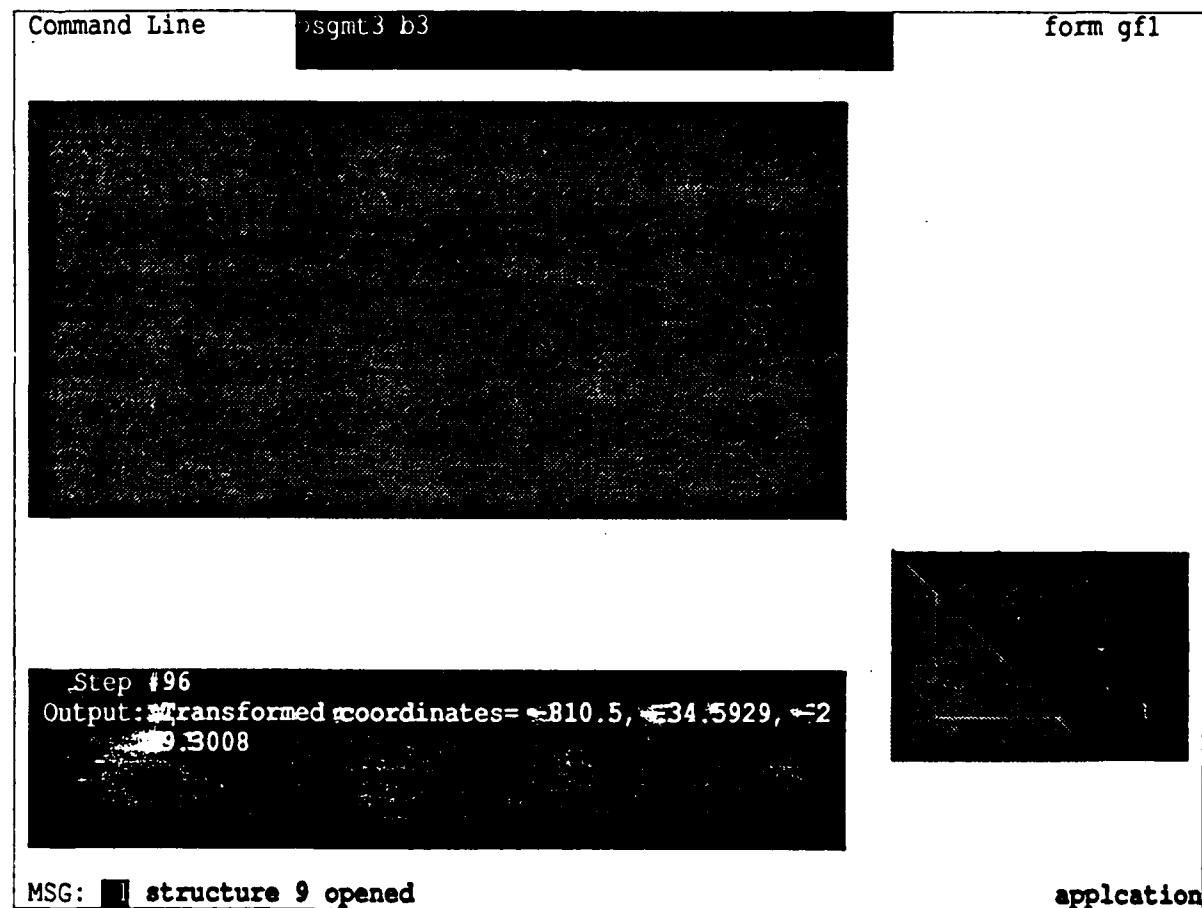
Figure 5-98a   (BEFORE)

Figure 5-98b   (AFTER)

```
Command Line          clst                                     form gf1
```

Step #97
Output: Transformed coordinates= -810.5, -34.5929, -2
       9.3008

MSG: 1 3-D transformation matrix element inserted   applcation
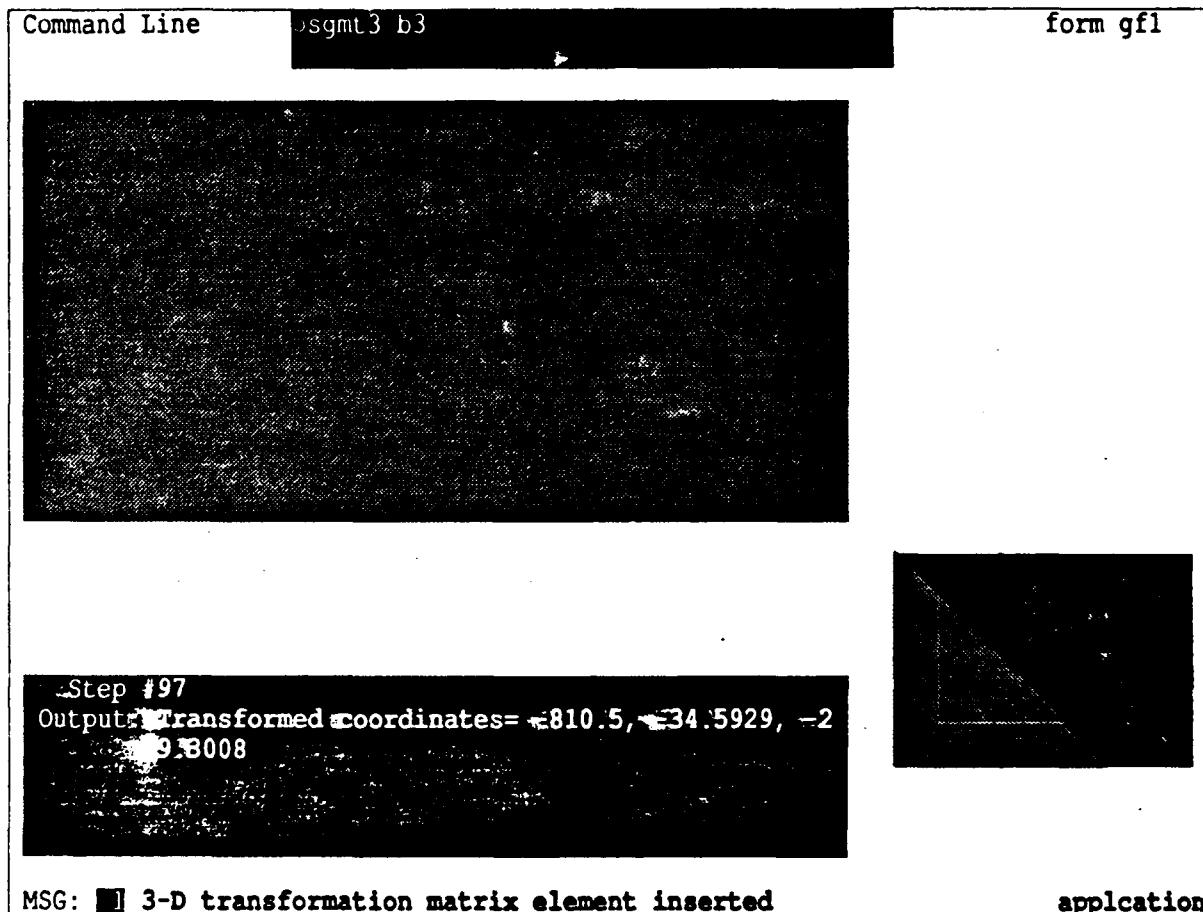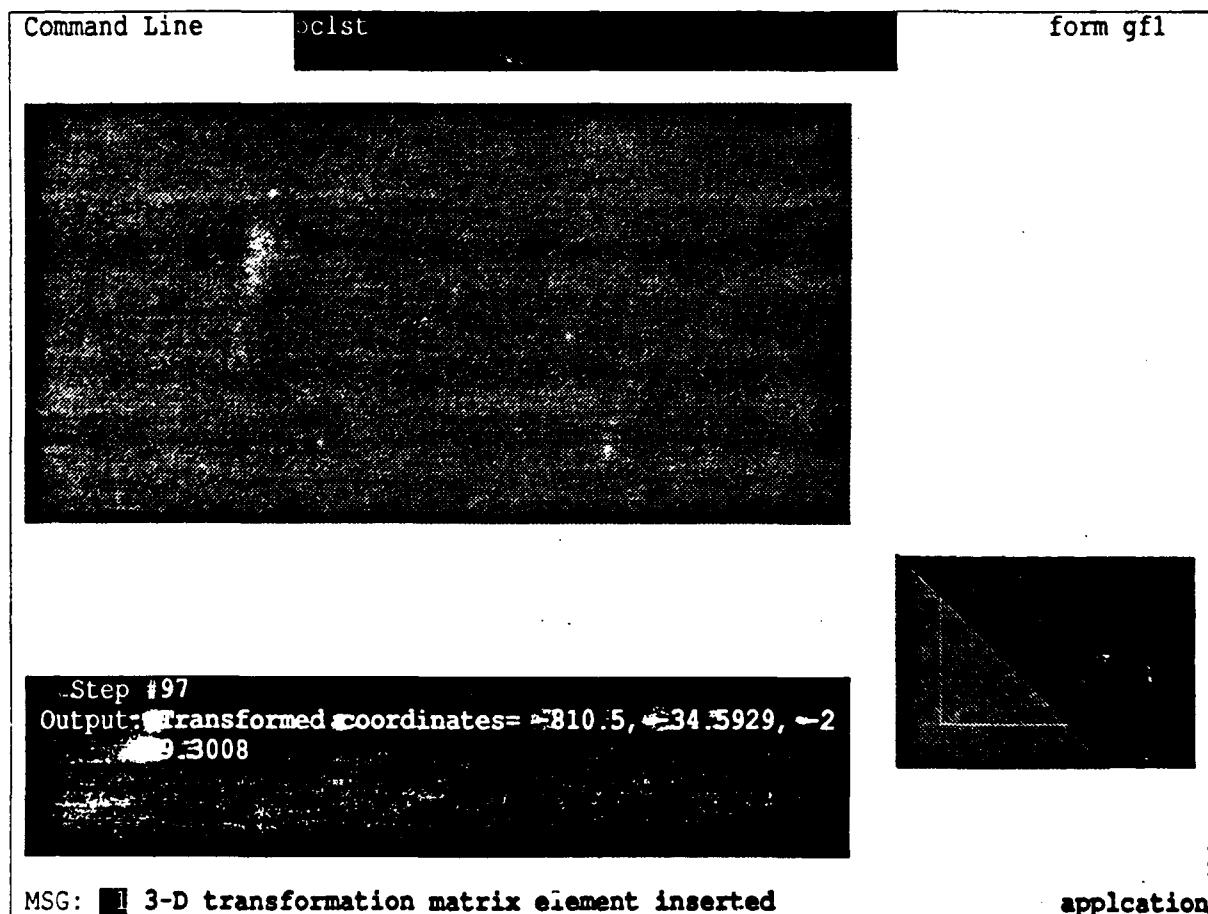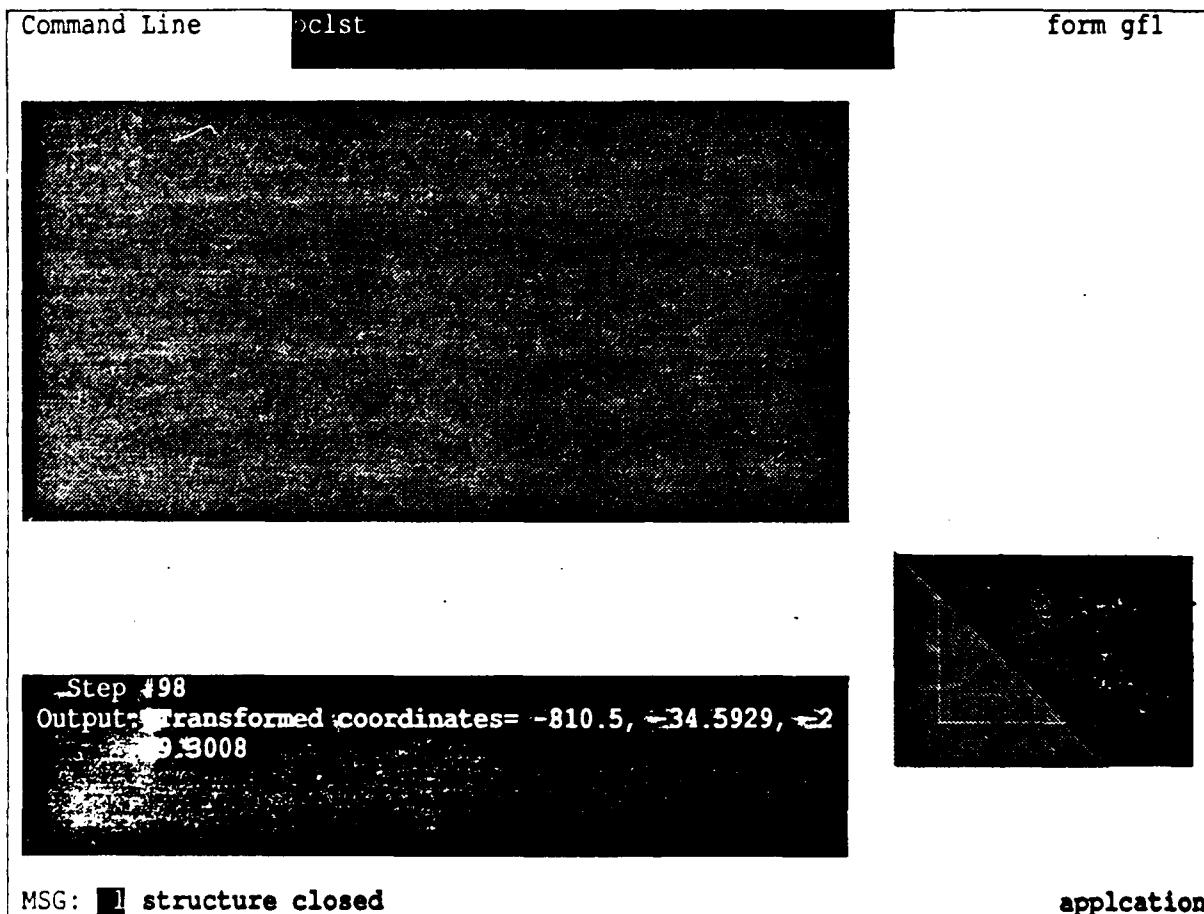
Figure 5-99a   (BEFORE)

Figure 5-99b (AFTER)

Command Line   ɔpost 1 9 1   form gf1

Step #98
Output: Transformed coordinates= -810.5, -34.5929, -2
9.3008
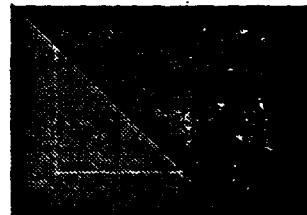
MSG: structure closed   applcation

Figure 5-100a   (BEFORE)
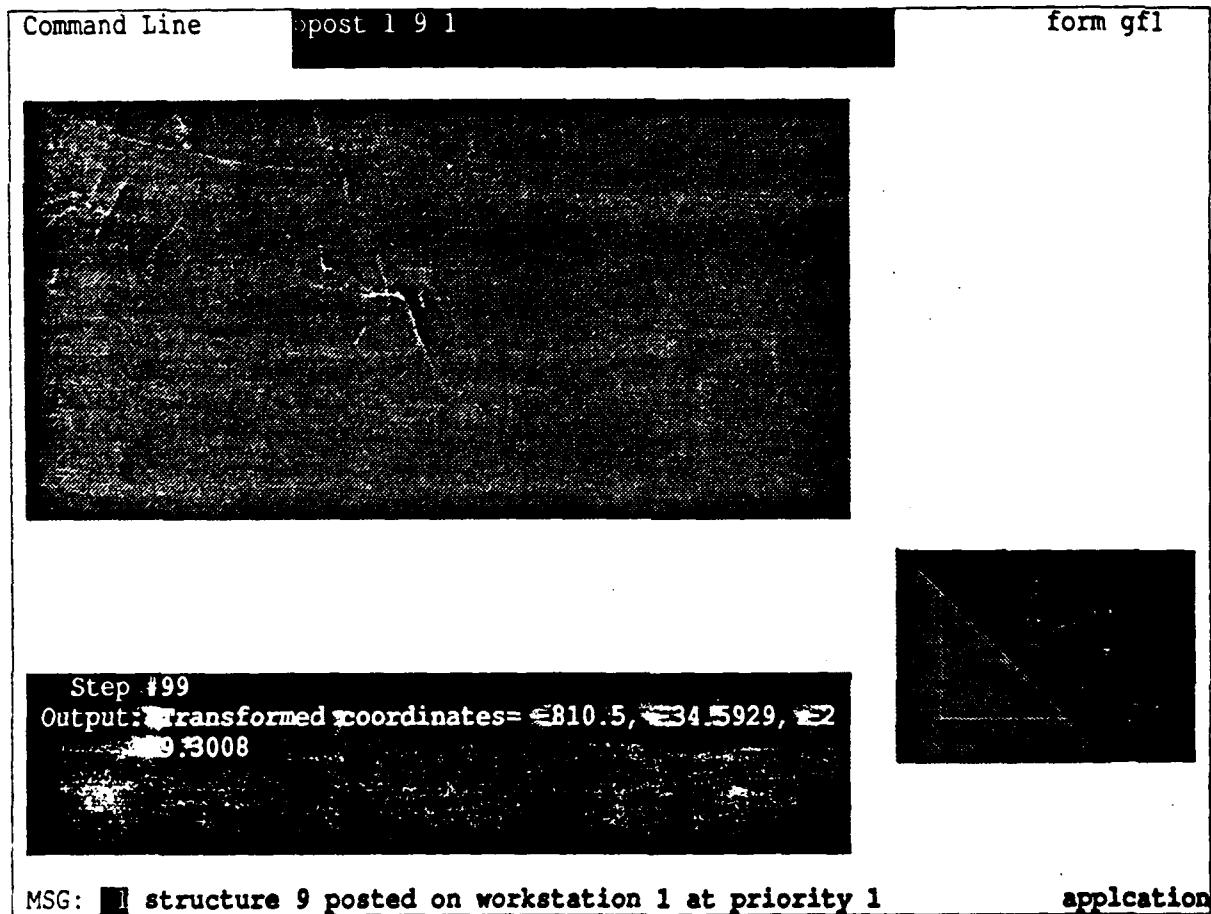
Figure 5-100b   (AFTER)
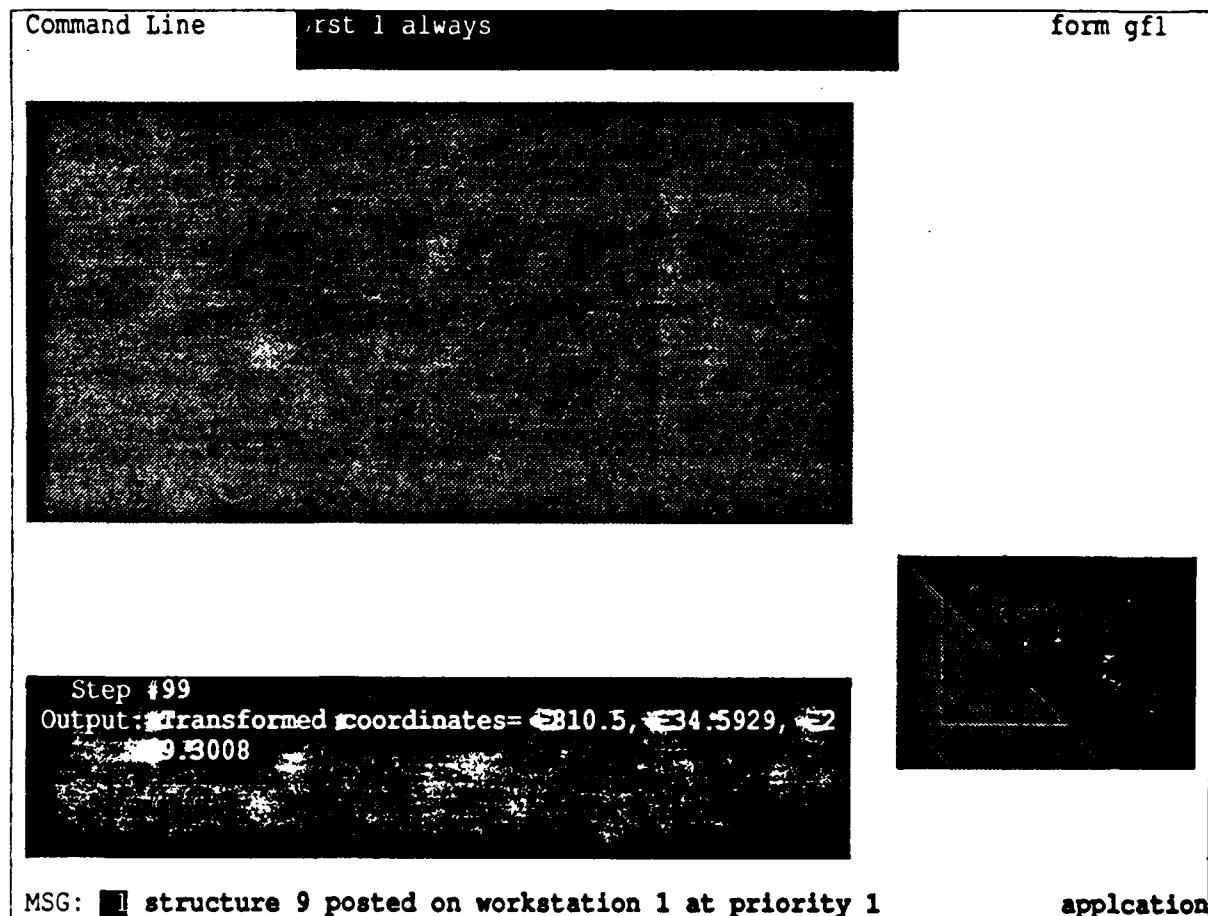
Figure 5-101a   (BEFORE)
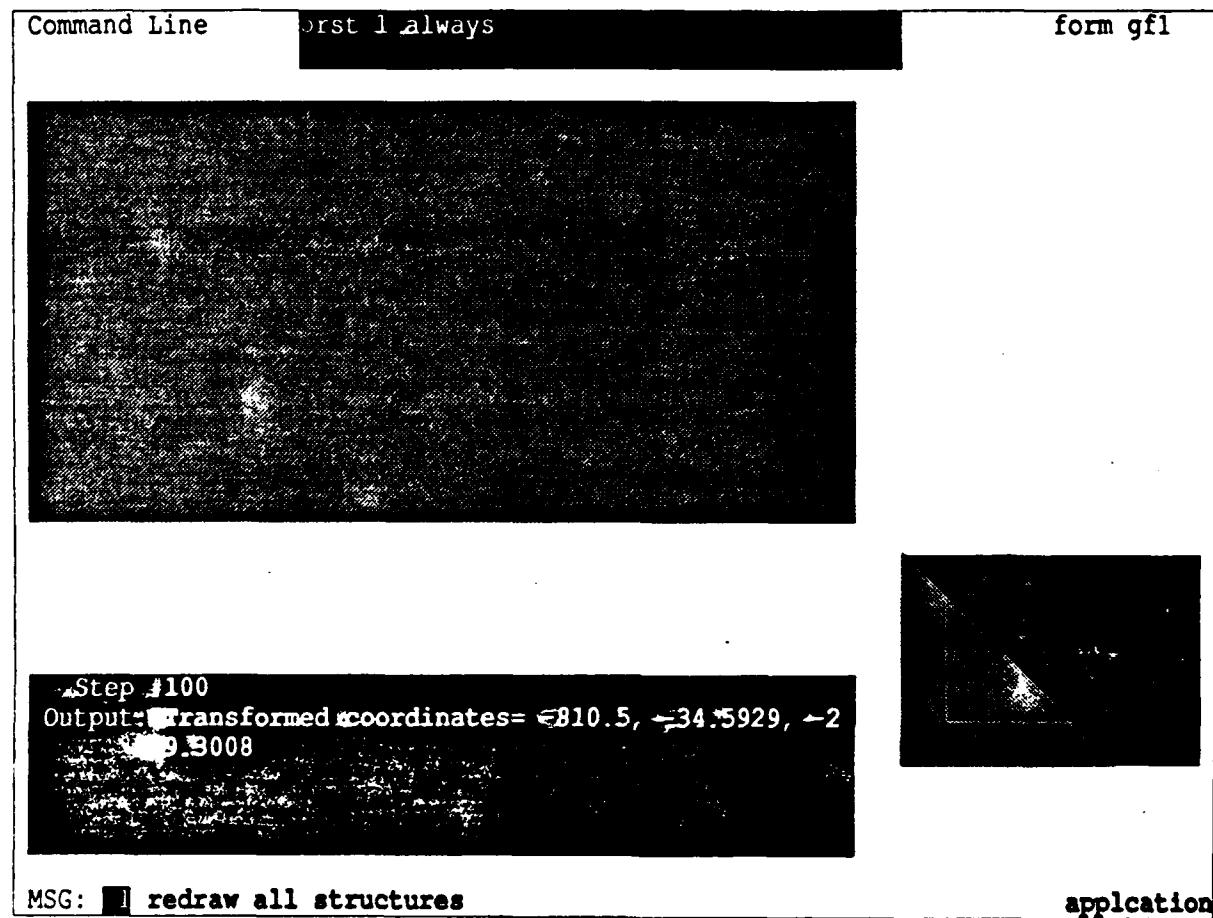
Figure 5-101b   (AFTER)
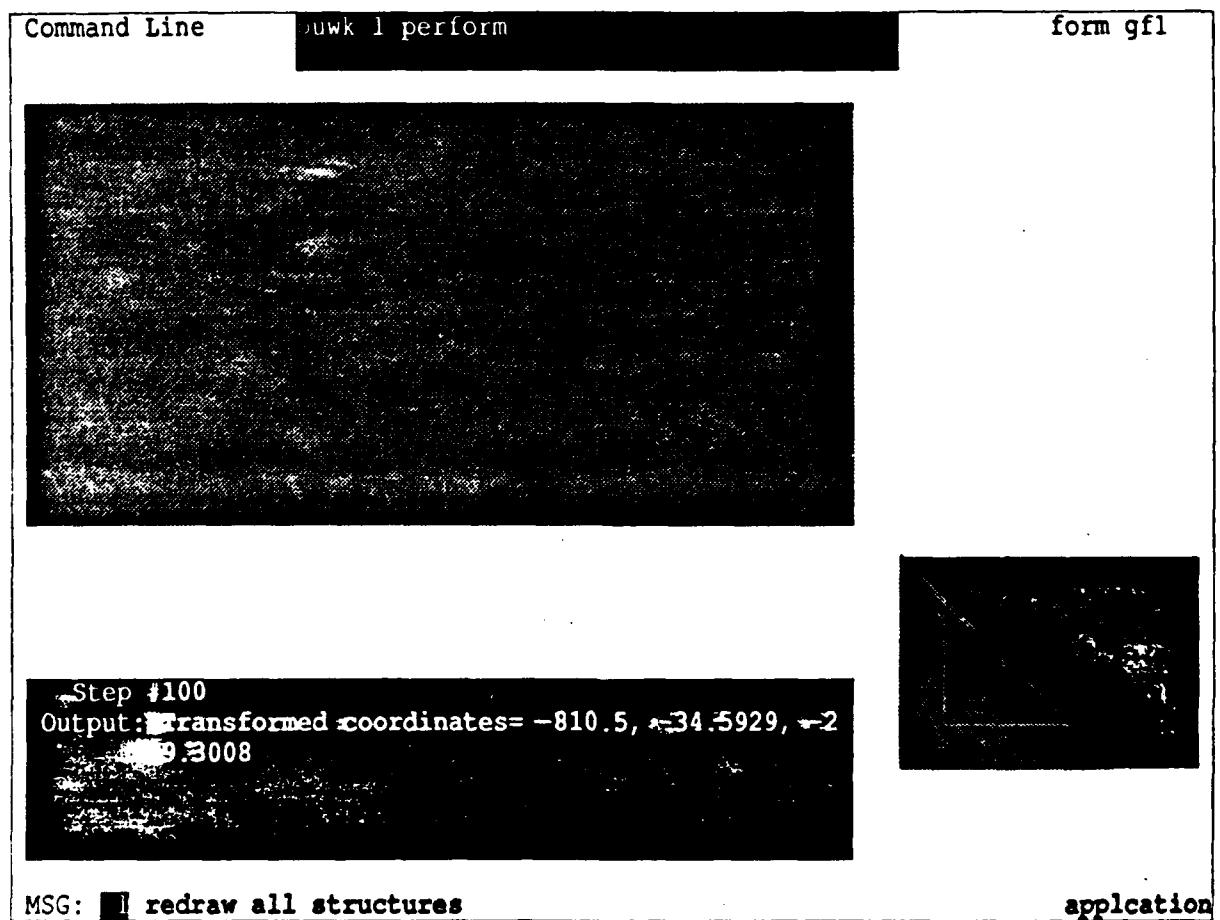
Figure 5-102a   (BEFORE)
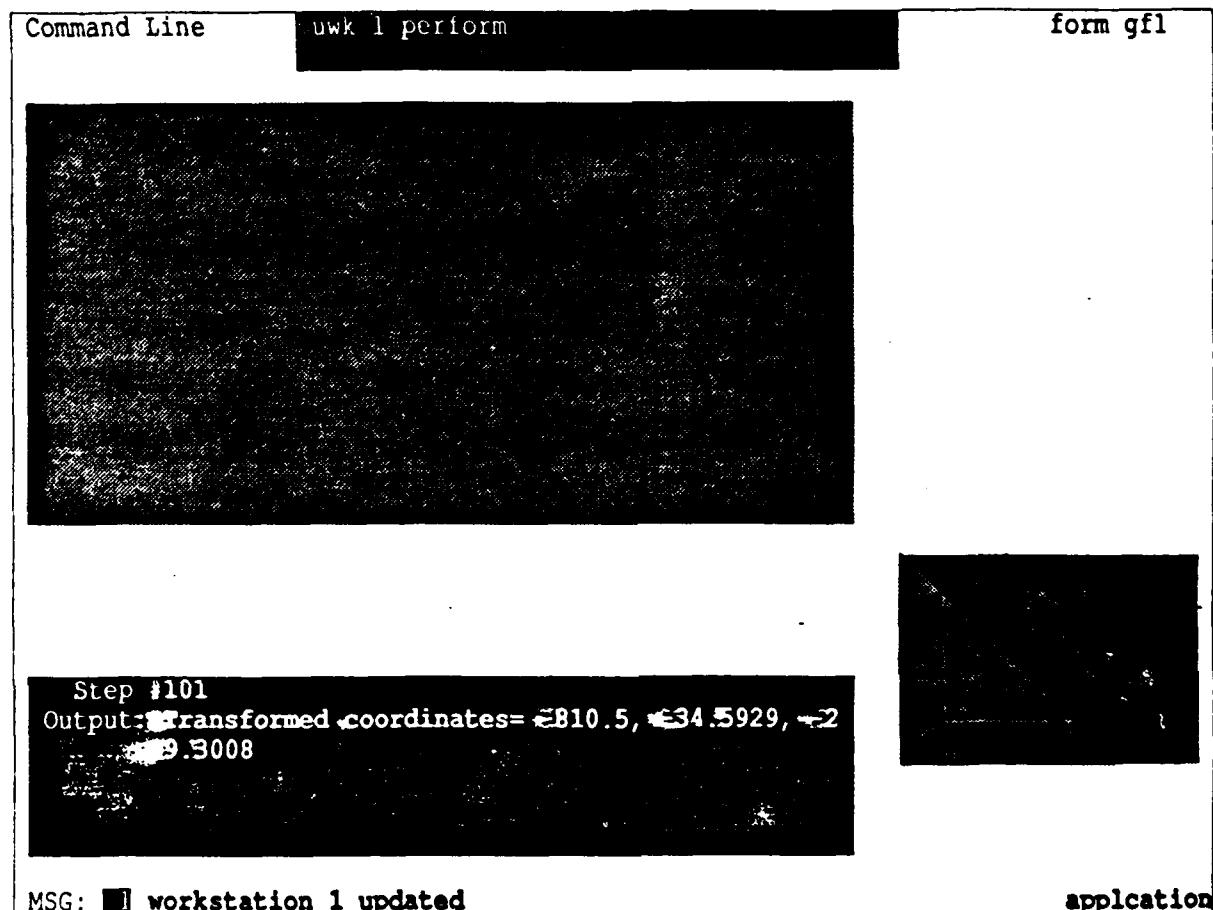
Command Line   uwk 1 perform   form gf1

Step #101
Output: Transformed coordinates= 810.5, 34.5929, -2
.3008

MSG: ▮ workstation 1 updated   applcation

Figure 5-102b   (AFTER)

Command Line   msg "GSS UTP output is complete"   form gfl

Step #101
Output: Transformed coordinates= -810.5, -34.5929, -2
9.3008

MSG: ☐ workstation 1 updated   application

Figure 5-103a   (BEFORE)

Figure 5-103b   (AFTER)

Figure 5-104a   (BEFORE)

Figure 5-104b   (AFTER)

Figure 5-105a   (BEFORE)

Command Line   ɔdas   form gf1

Step #103
Output: transformed coordinates= -810.5, -34.5929, -2
                9.8008

MSG: 1 all structures deleted   applcation

Figure 5-105b   (AFTER)

Figure 5-106a   (BEFORE)

Command Line   bqewk   form gf1

Step #104
Output: error and (return code) is 0
        total size required is 120
        types are in the message queue

MSG: ▮ FP WINDOWS   applcation

Figure 5-106b   (AFTER)

```
Command Line        qwkc l                                    form gf1
```
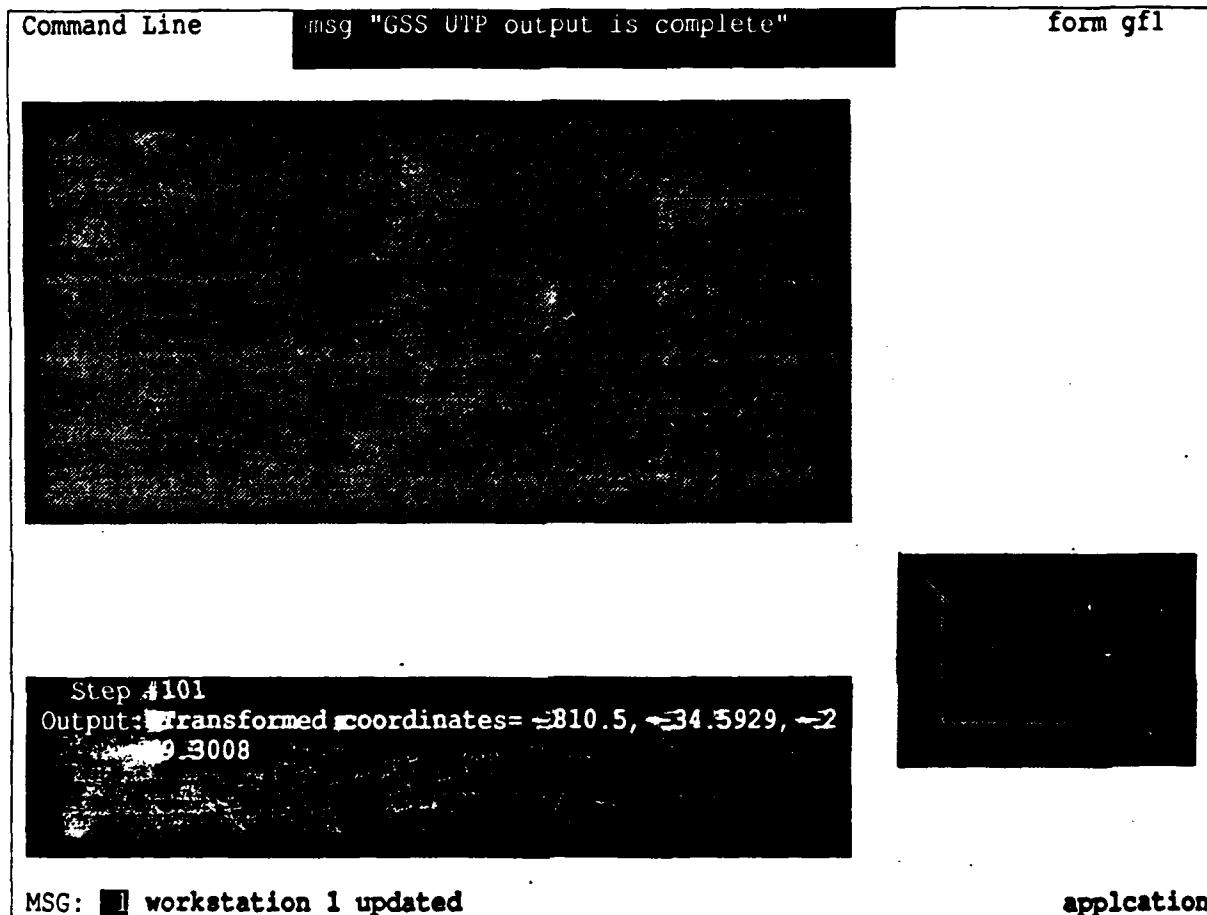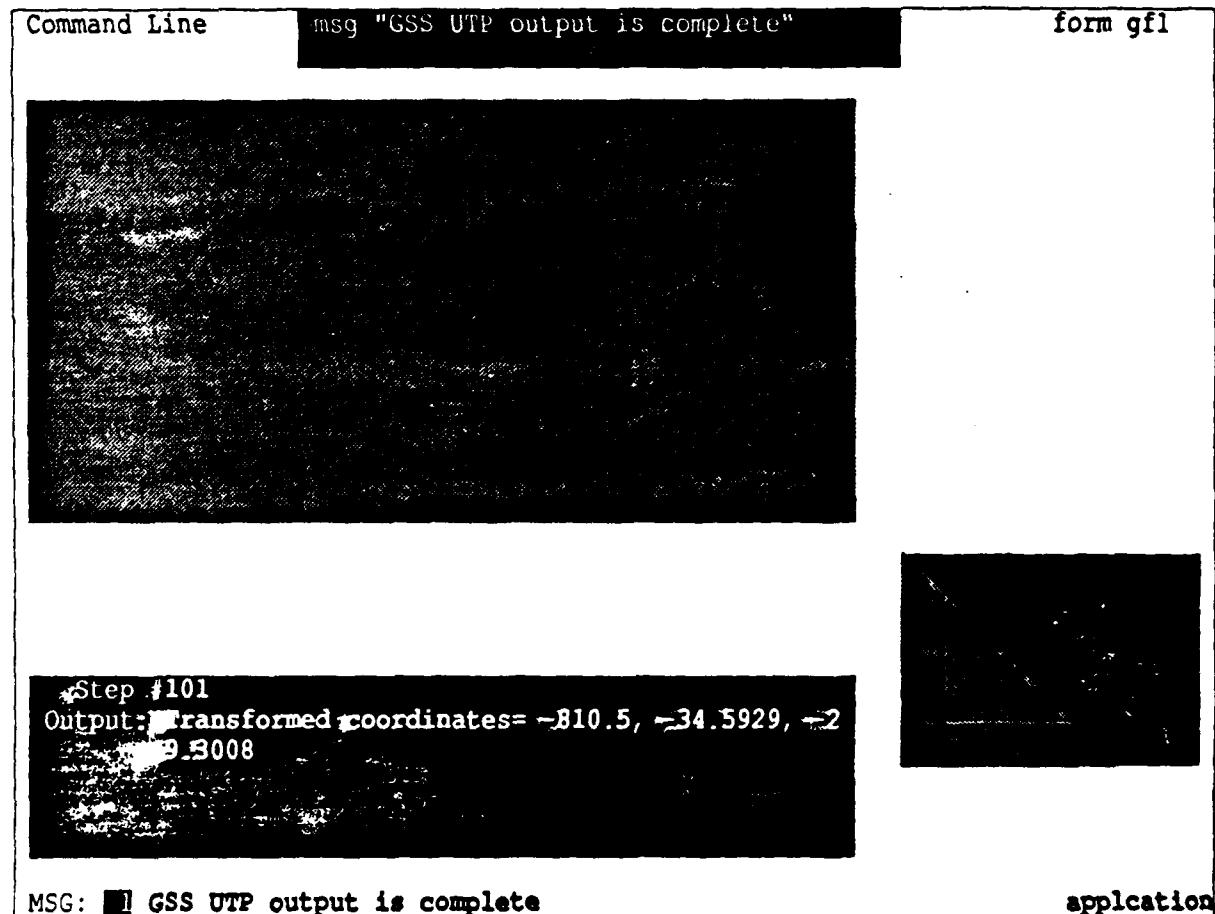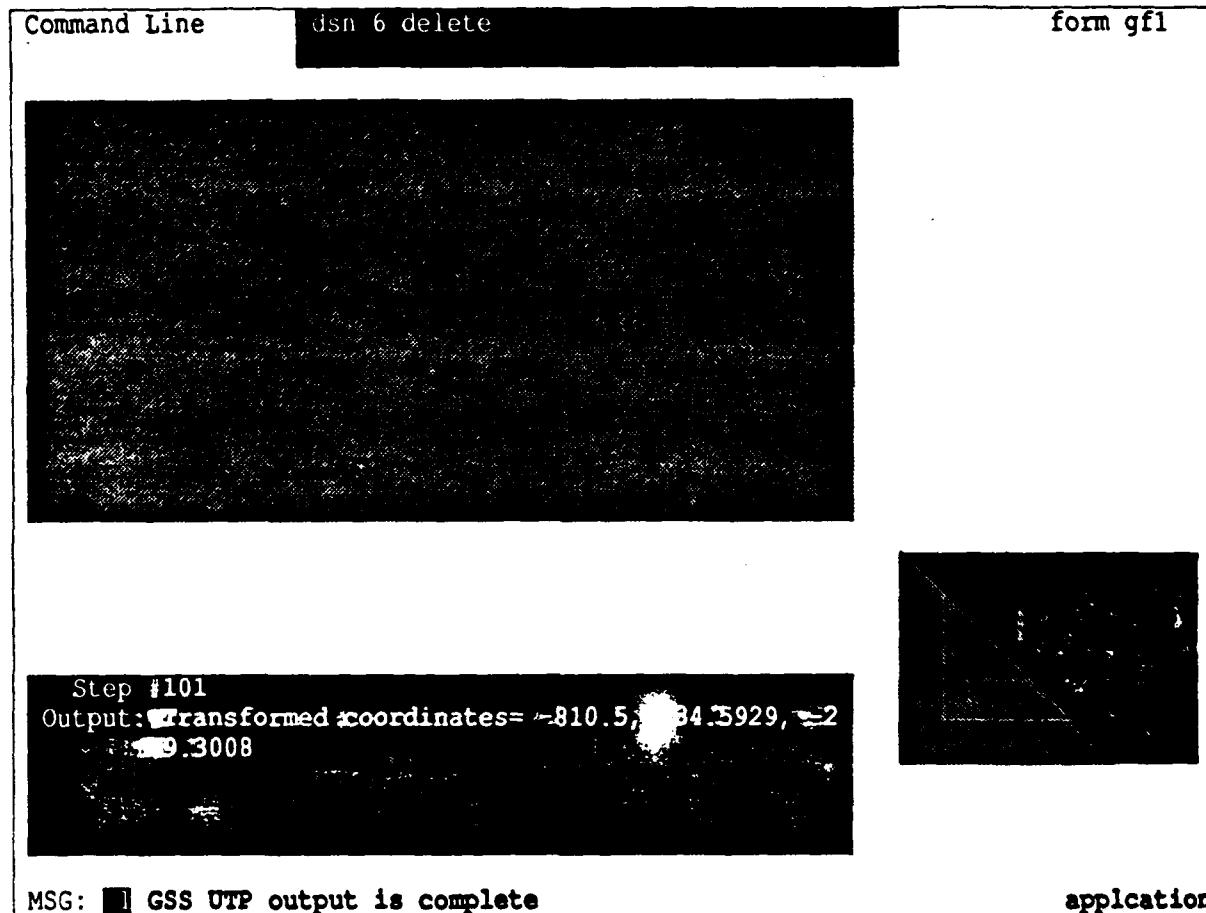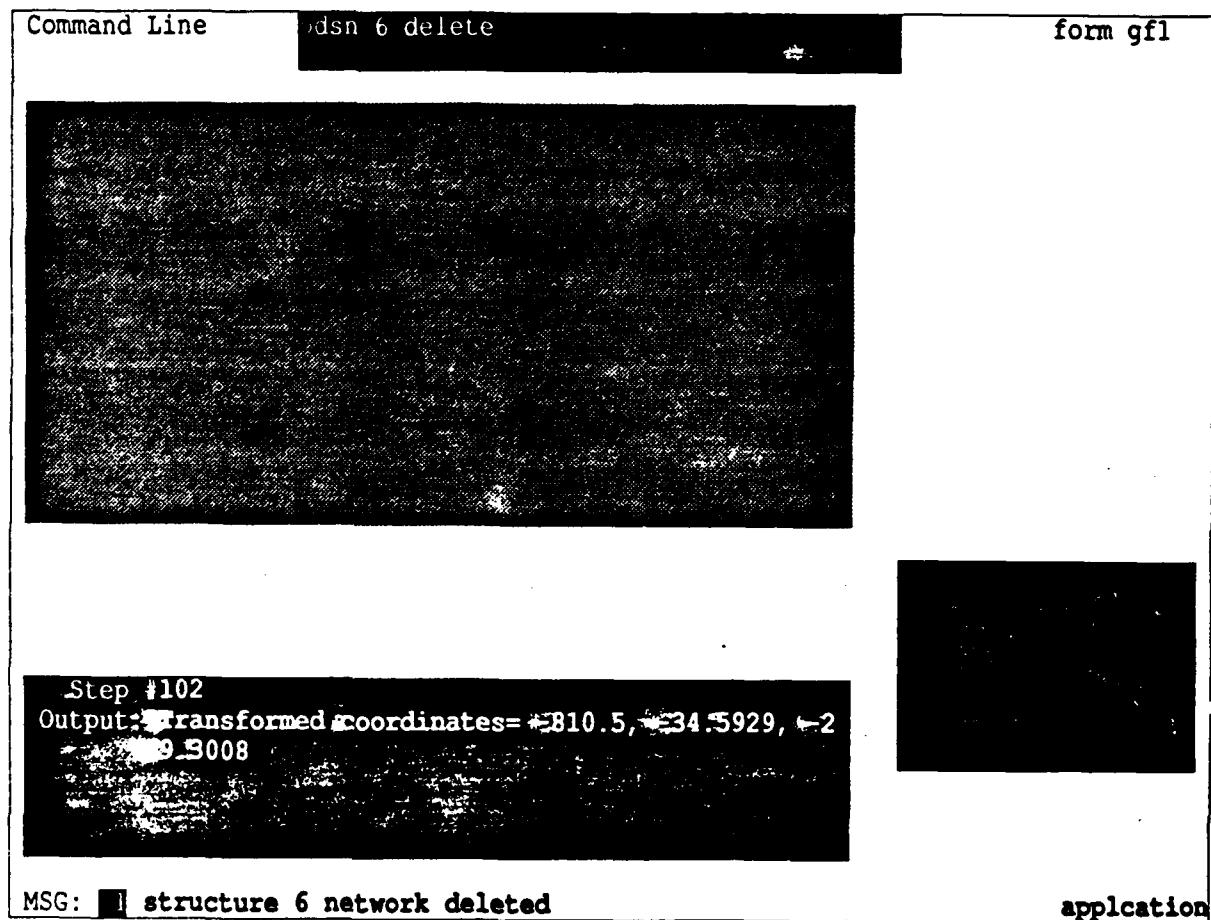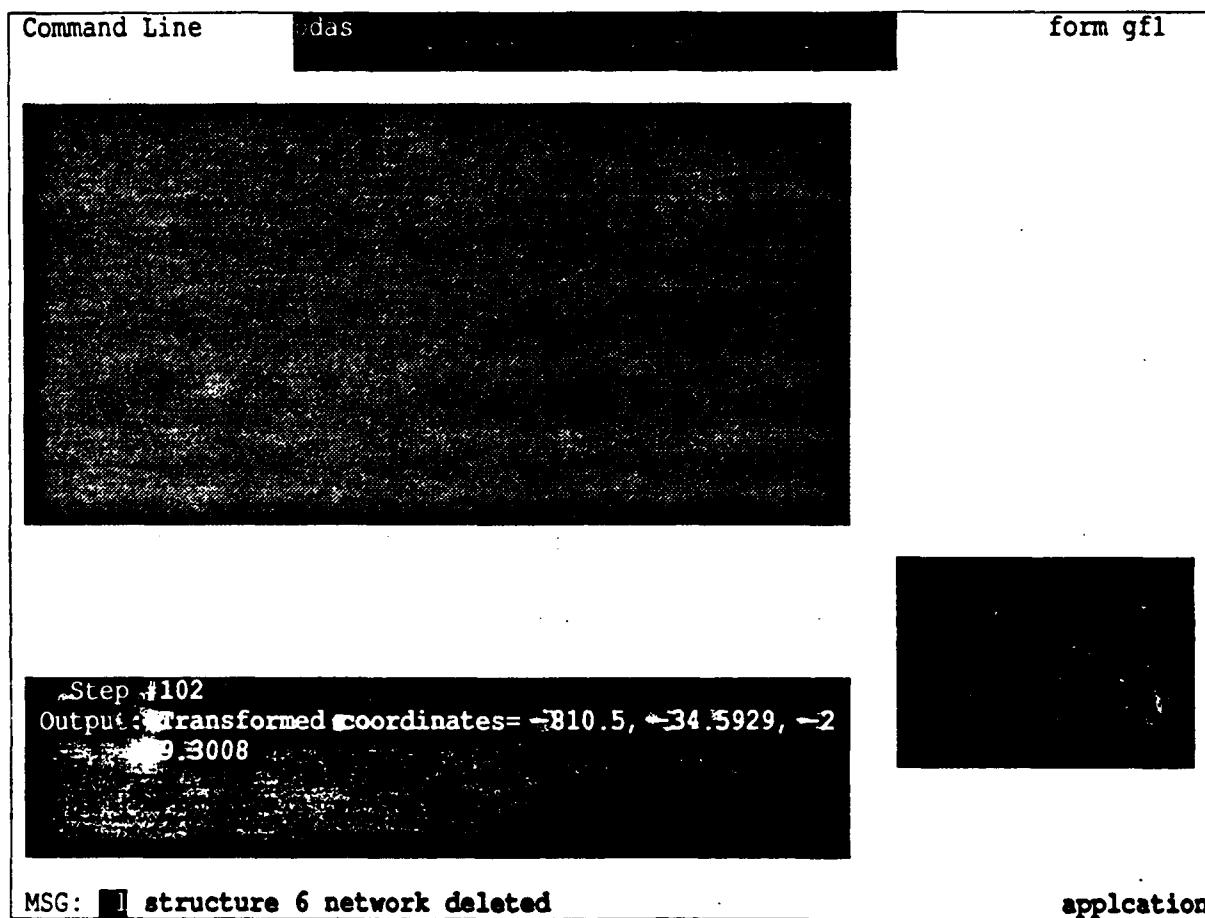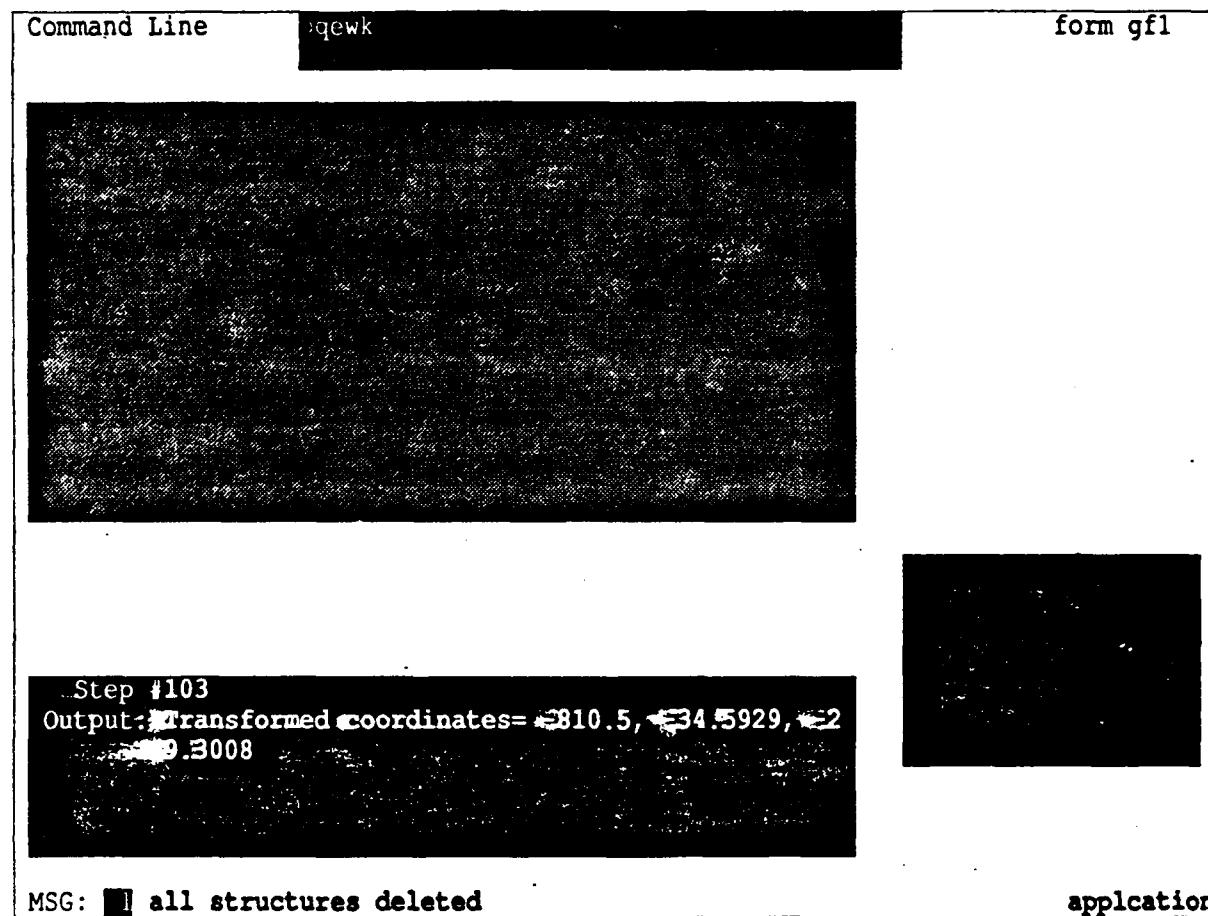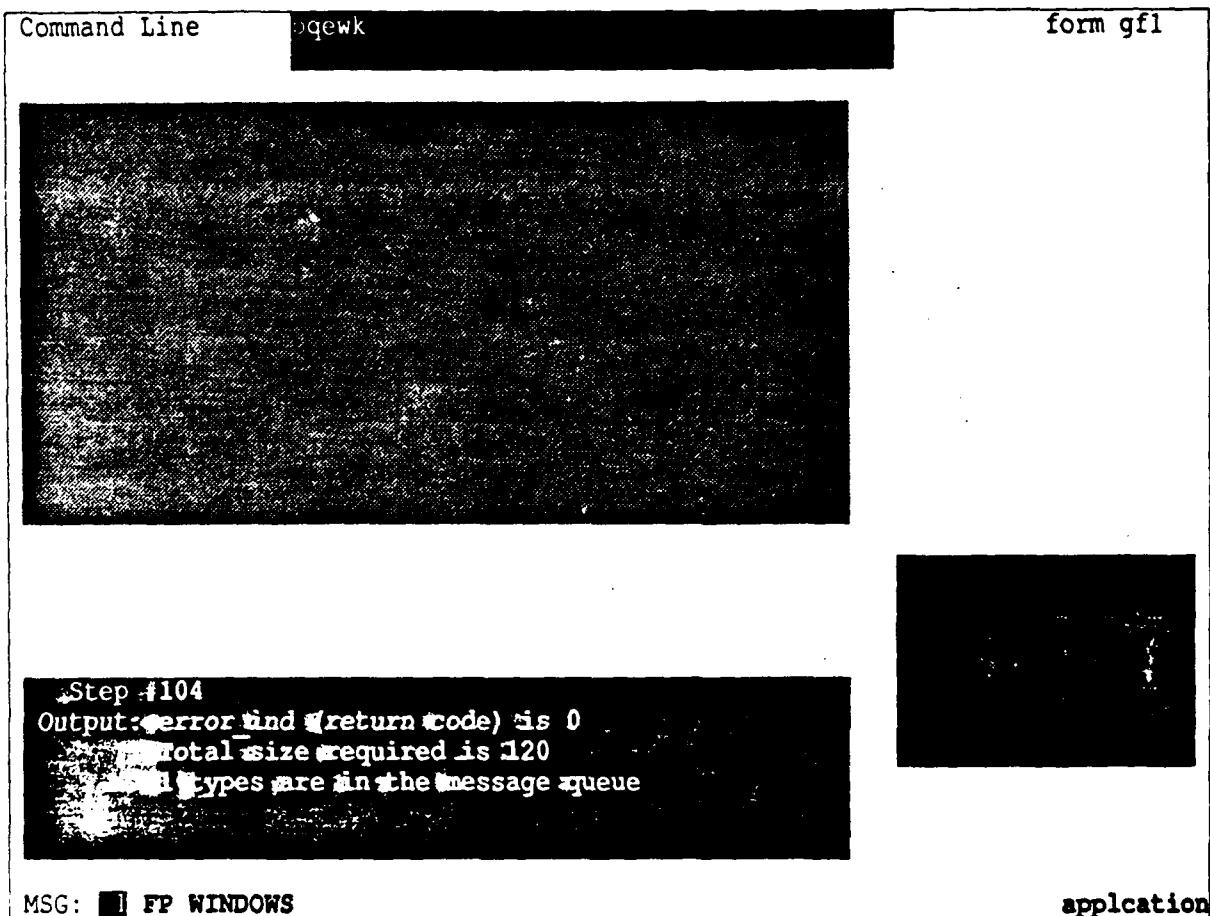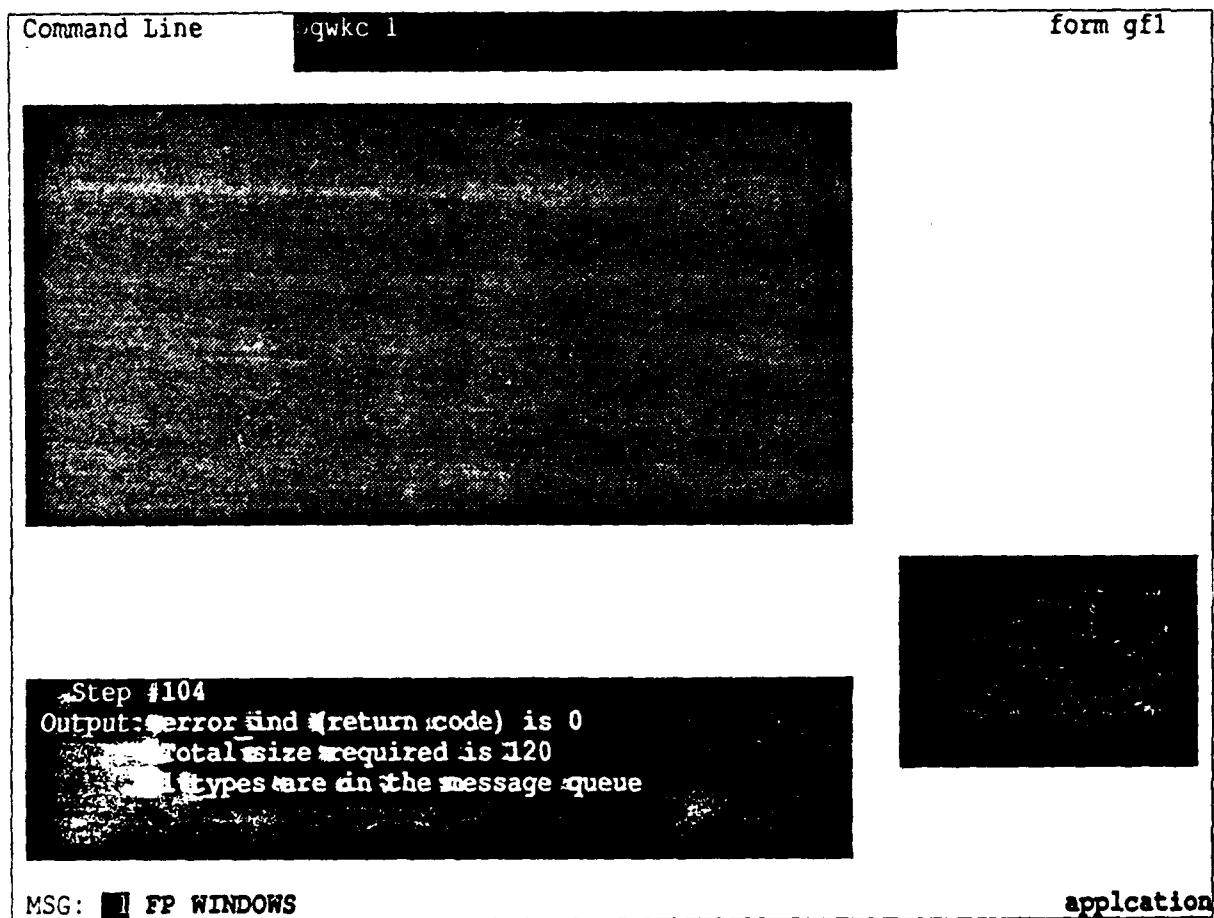
Figure 5-107a   (BEFORE)
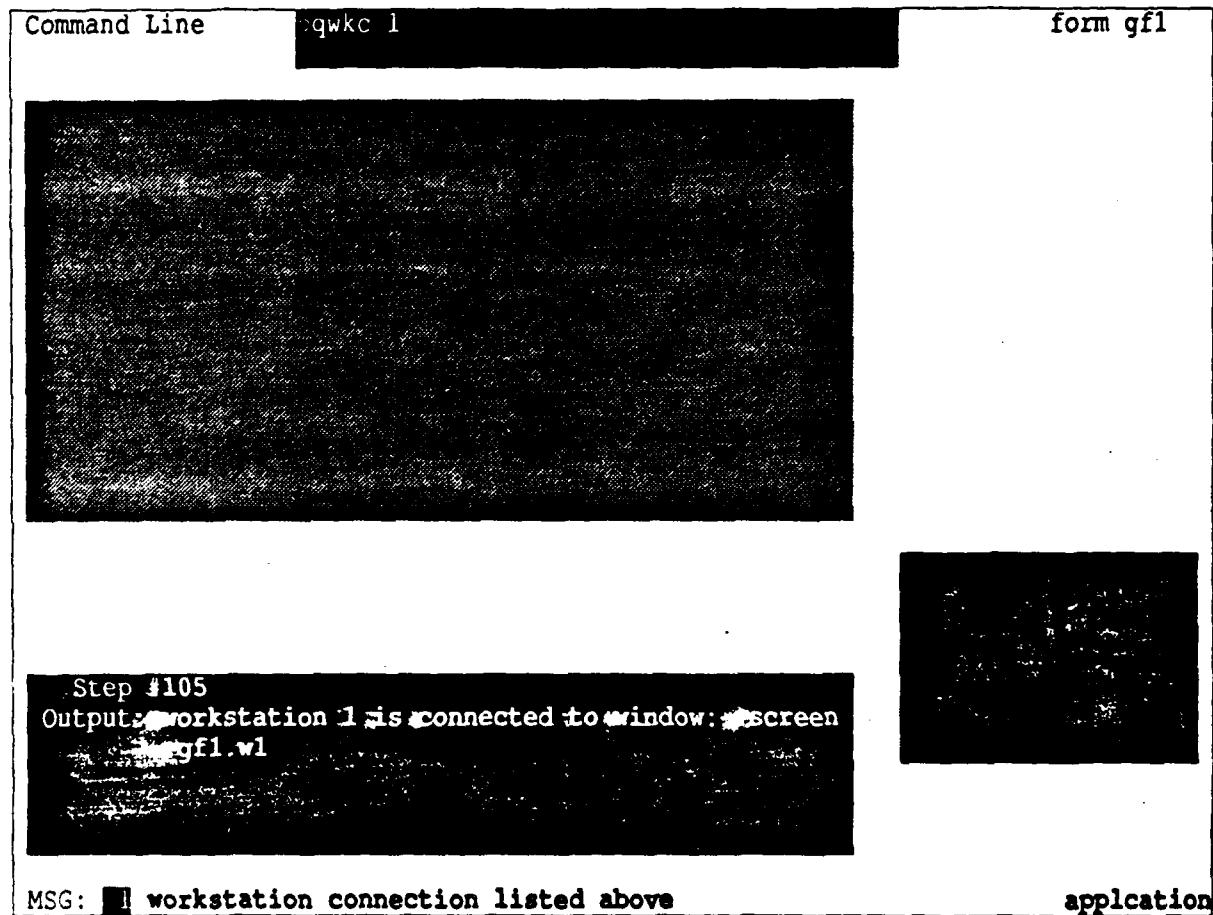
Figure 5-107b   (AFTER)

Figure 5-108a   (BEFORE)
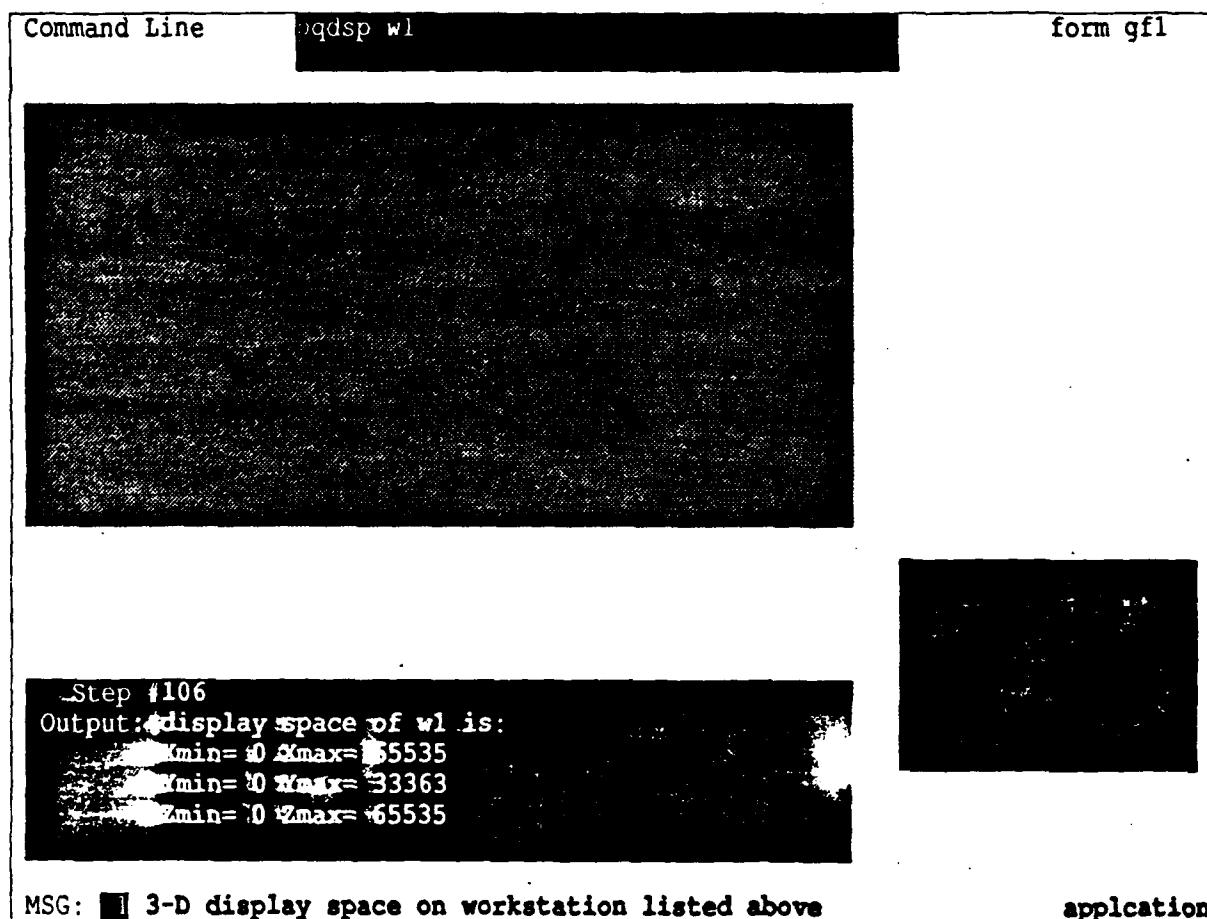
Command Line   ,qdsp3 w1   form gf1

Step #106
Output: display space of w1 is:
        Xmin= 0 Xmax= 65535
        Ymin= 0 Ymax= 33363
        Zmin= 0 Zmax= 65535

MSG: ■ 3-D display space on workstation listed above   applcation

Figure 5-108b   (AFTER)

Command Line   )qdsp wl   form gfl

Step #106
Output: display space of wl is:
    Xmin= 0 Xmax= 65535
    Ymin= 0 Ymax= 33363
    Zmin= 0 Zmax= 65535

MSG: ▮ 3-D display space on workstation listed above   applcation
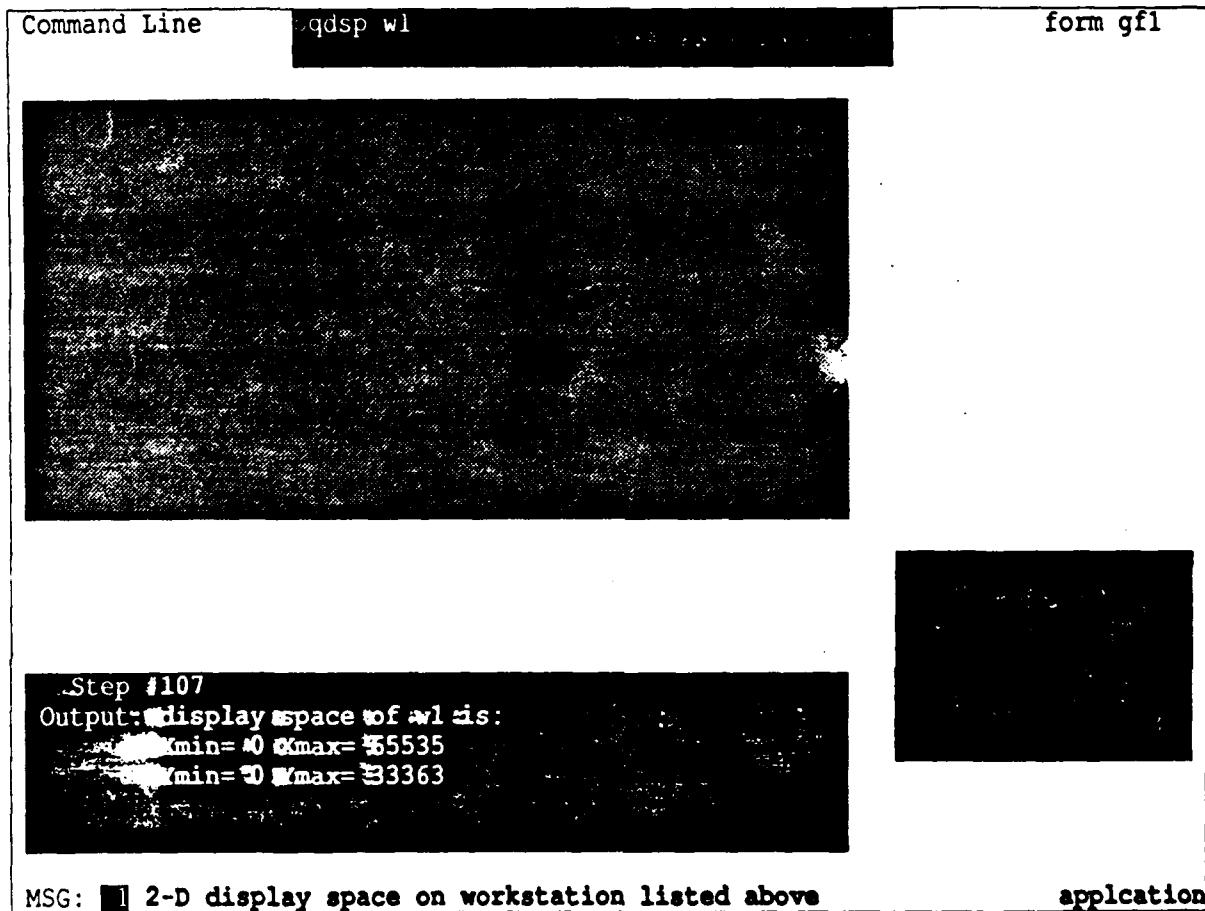
Figure 5-109a   (BEFORE)
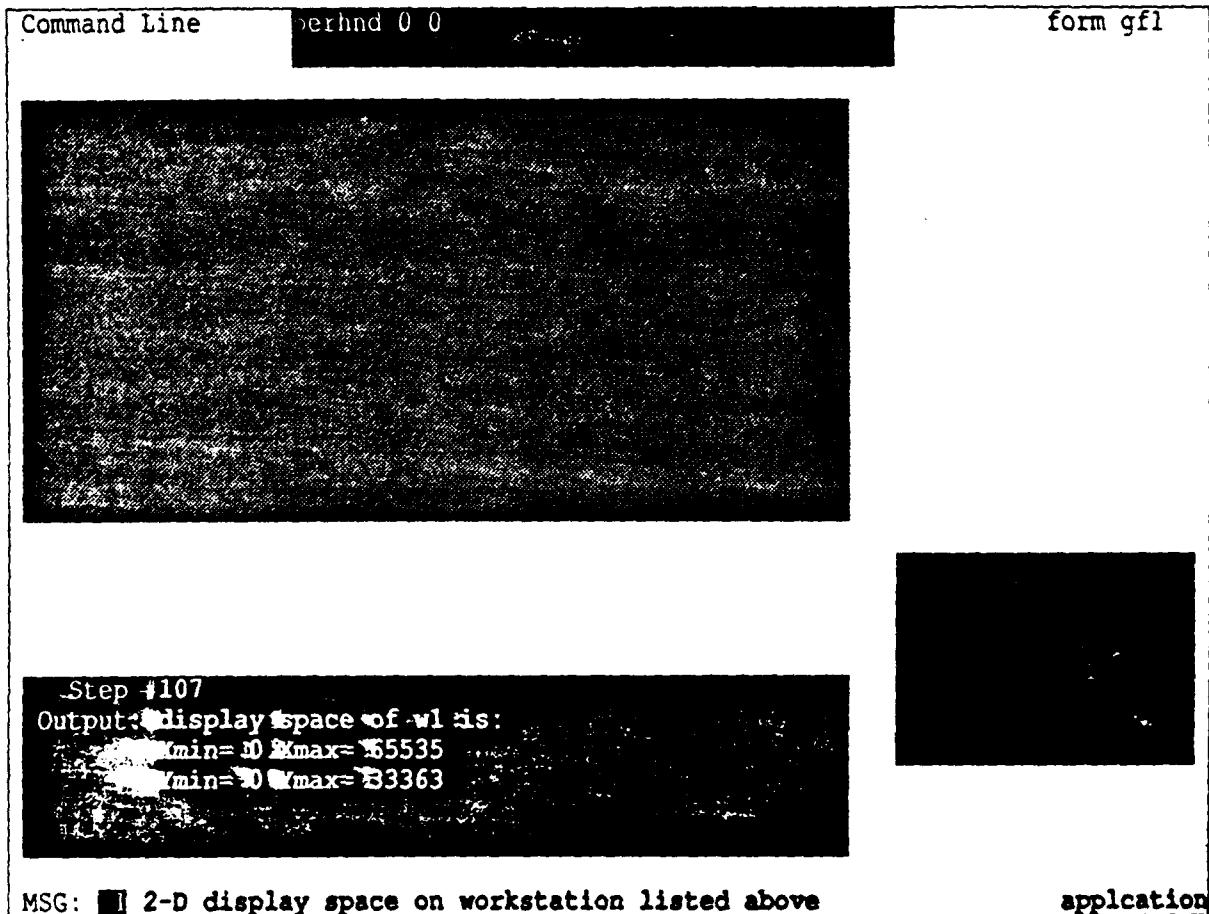
Figure 5-109b   (AFTER)

Figure 5-109c   (Message Queue, AFTER)
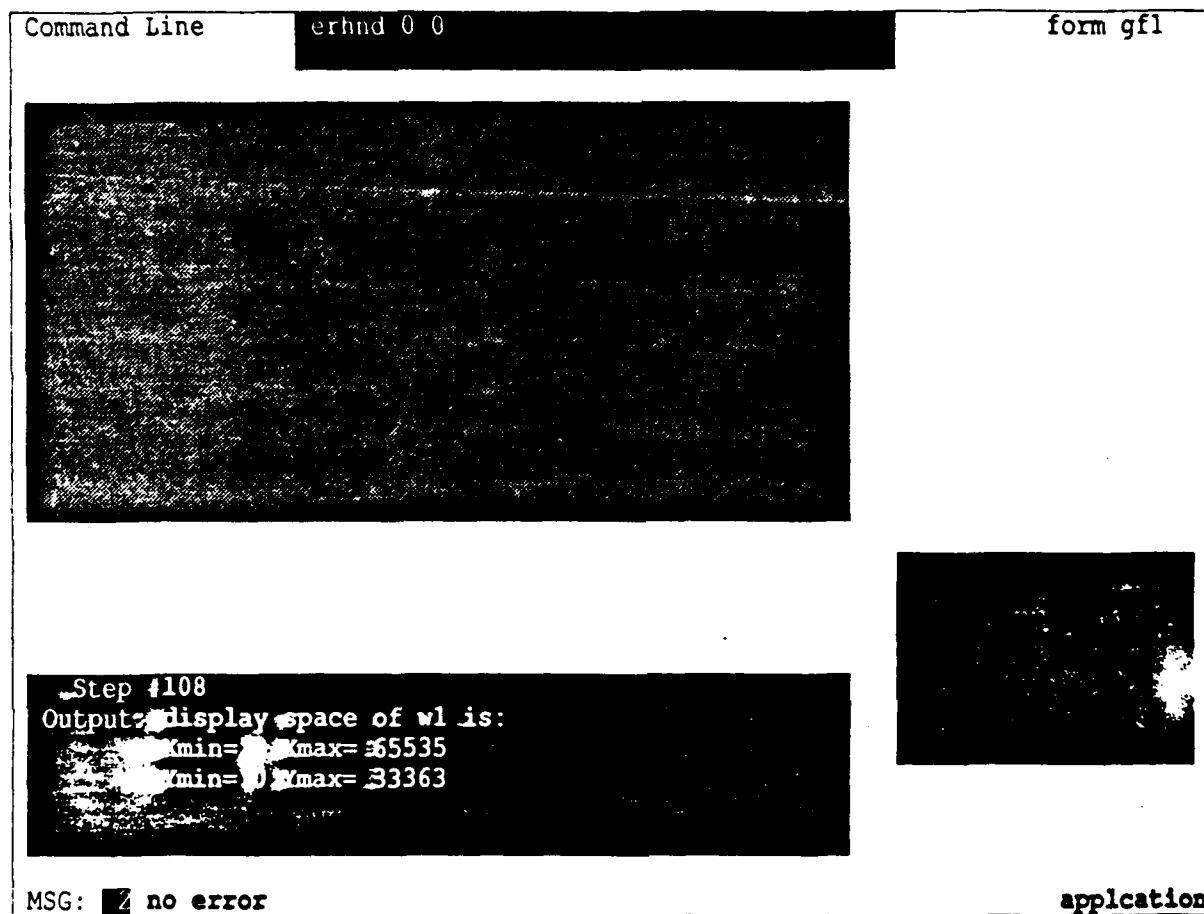
Command Line   erhnd 0 0   form gfl

Step #108
Output: display space of wl is:
       Xmin=   Xmax= 65535
       Ymin= 0 Ymax= 33363

MSG: no error   applcation

Figure 5-110a   (BEFORE)

```
┌──────────────────────────────────────────────────────────────┐
│                        Message Queue                           │
│                                                                │
│ Msg: ▓ no error                                                │
│ Msg: █ PHIGS error 0000 from routine UNKNOWN is                │
│                                                                │
│                                                                │
│                                                                │
│                                                                │
│                                                                │
│                                                                │
│                                                                │
│                                                                │
│                                                                │
│                                                                │
│                                                                │
│                                                                │
│                                                                │
│                                                                │
│                                                                │
│                                                                │
│ MSG: ▓ no error                                     applcation │
└──────────────────────────────────────────────────────────────┘
```

Figure 5-110b   (AFTER)

Command Line   erlog 0 0   form gf1

Step #108
Output: display space of wl is:
    Xmin= 0 Xmax= 65535
    Ymin= 0 Ymax= 33363

MSG: no error   applcation
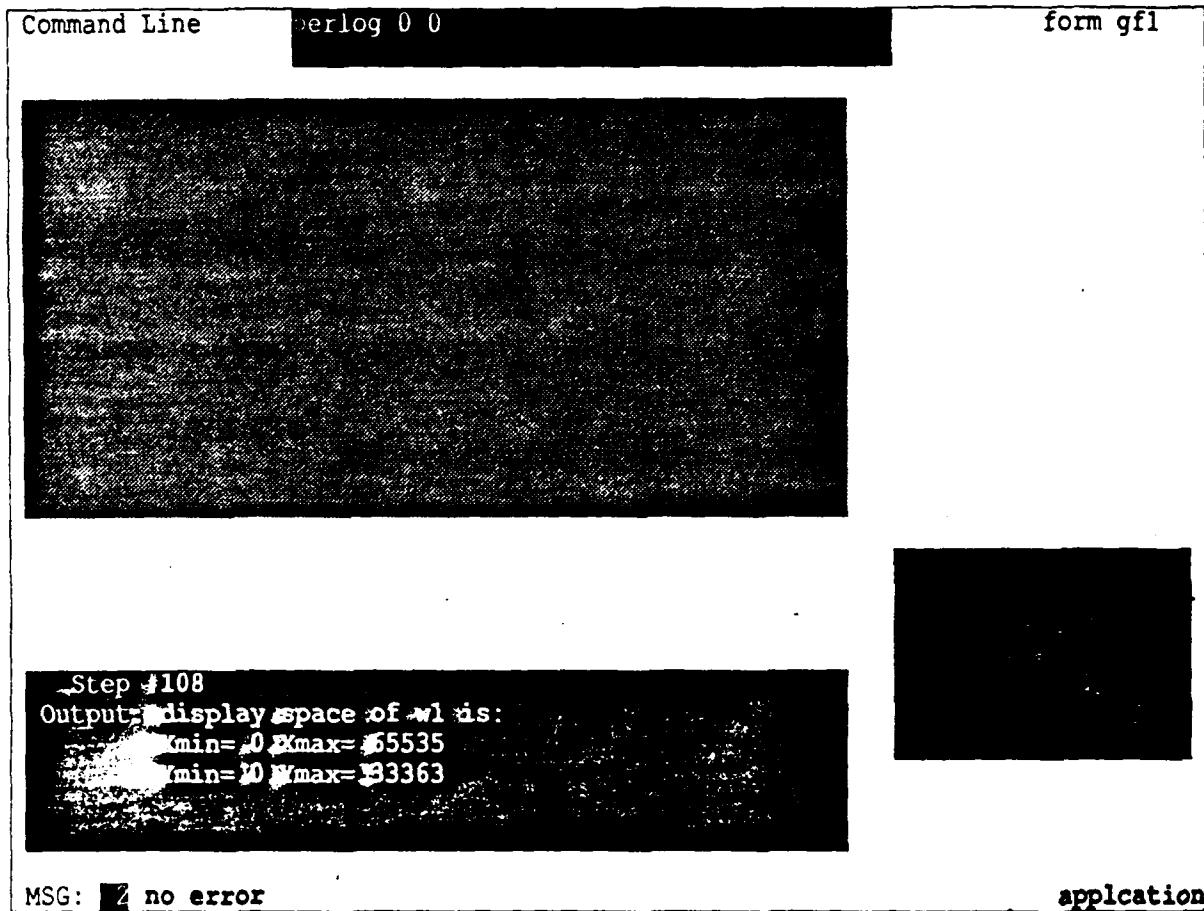
Figure 5-110c (Message Queue, AFTER)

Command Line   erlog 0 0   form gfl

Step #109
Output: display space of wl is:
          Xmin= 0 Xmax= 65535
          Ymin= 0 Ymax= 33363

MSG: 2 no error   applcation
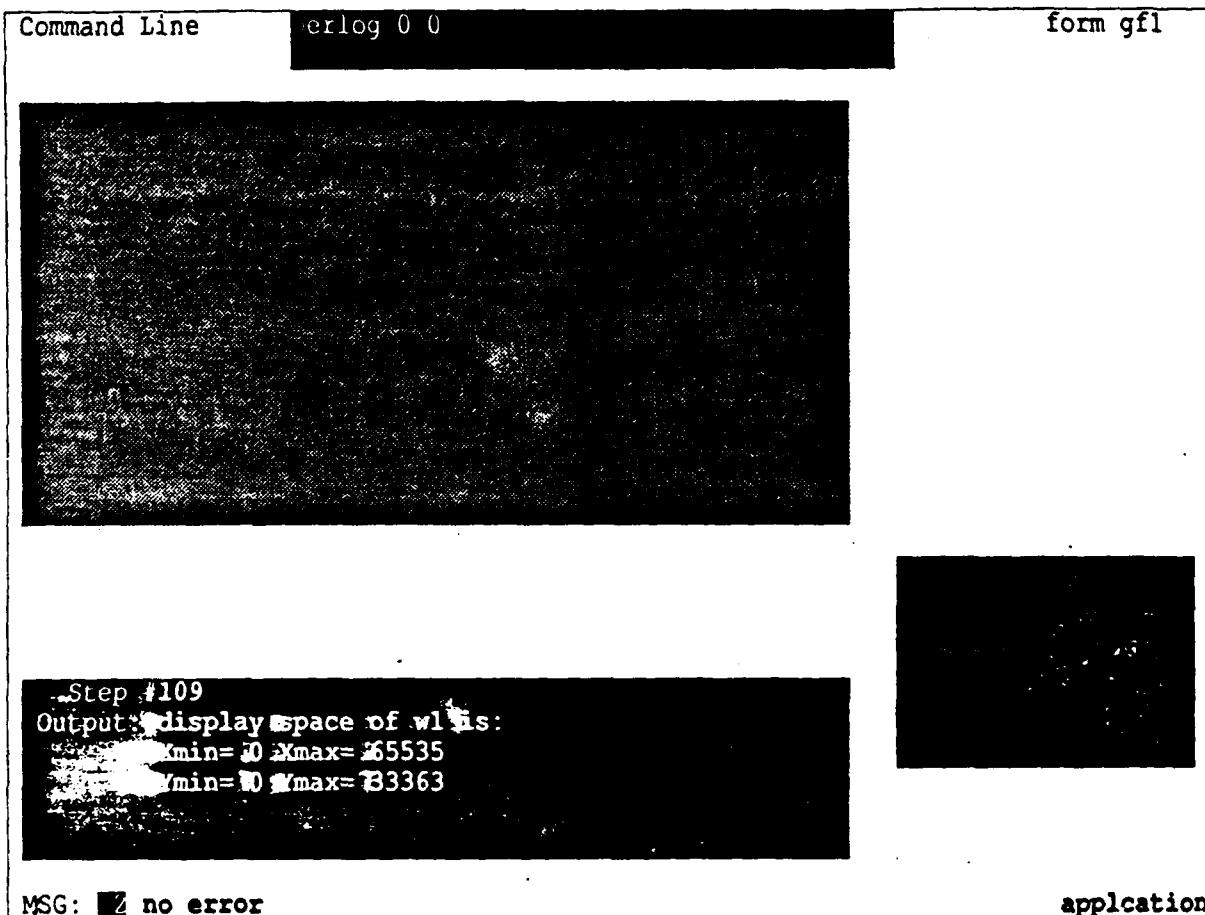
Figure 5-111a   (BEFORE)

```
+------------------------------------------------------------------+
|                         Message Queue                            |
|                                                                  |
| Msg: ▓ no error                                                  |
| Msg: ▓ PHIGS error 0000 from routine UNKNOWN is                  |
|                                                                  |
|                                                                  |
|                                                                  |
|                                                                  |
|                                                                  |
|                                                                  |
|                                                                  |
|                                                                  |
|                                                                  |
|                                                                  |
|                                                                  |
|                                                                  |
|                                                                  |
|                                                                  |
| MSG: ▓ no error                                      applcation  |
+------------------------------------------------------------------+
```

Figure 5-111b   (AFTER)
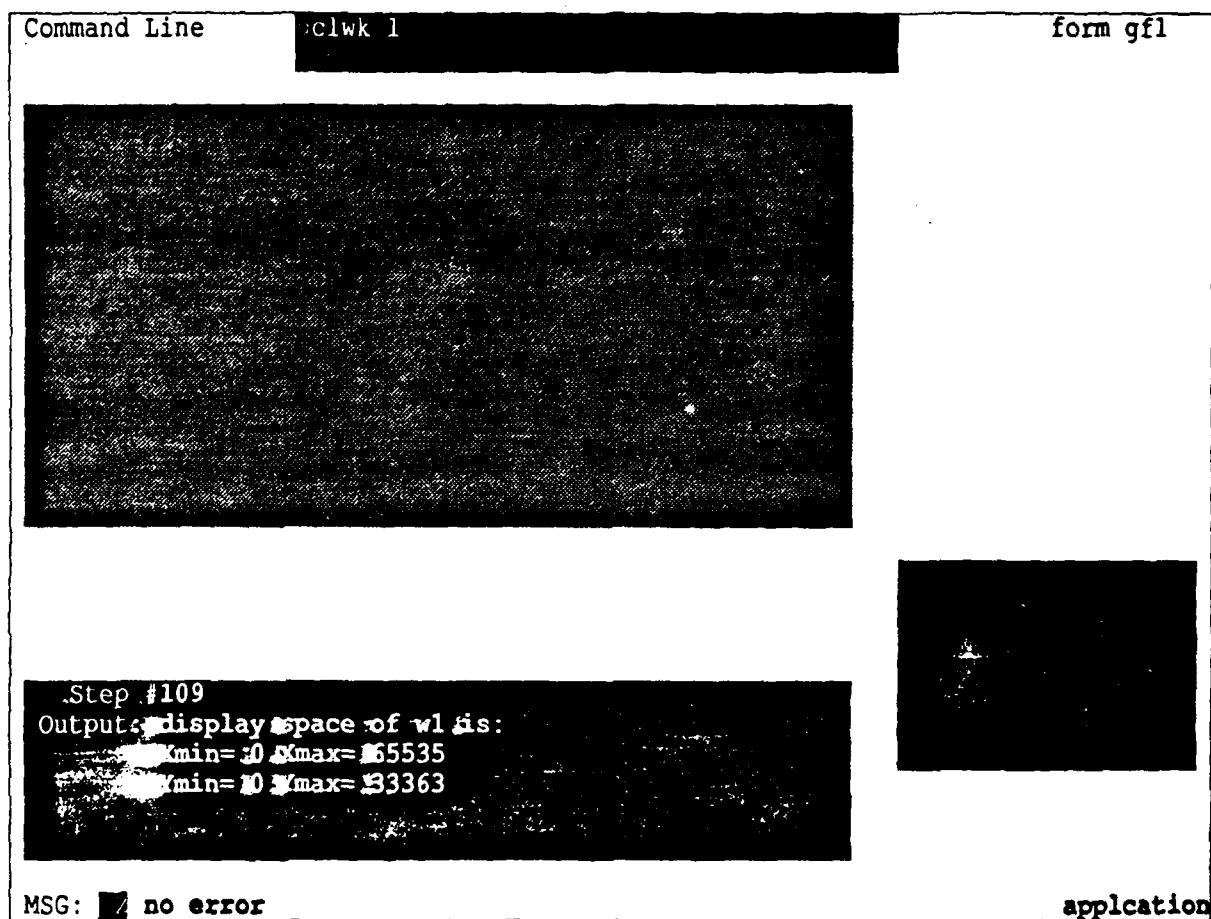
Figure 5-112a   (BEFORE)

Figure 5-112b   (AFTER)

Figure 5-113a   (BEFORE)

```
Command Line        clwk 2                              form gf1




Step #111
Output: display space of wl is:
        Xmin= 0 Xmax= 65535
        Ymin= 0 Ymax= 33363


MSG: ▓ workstation closed                            applcation
```
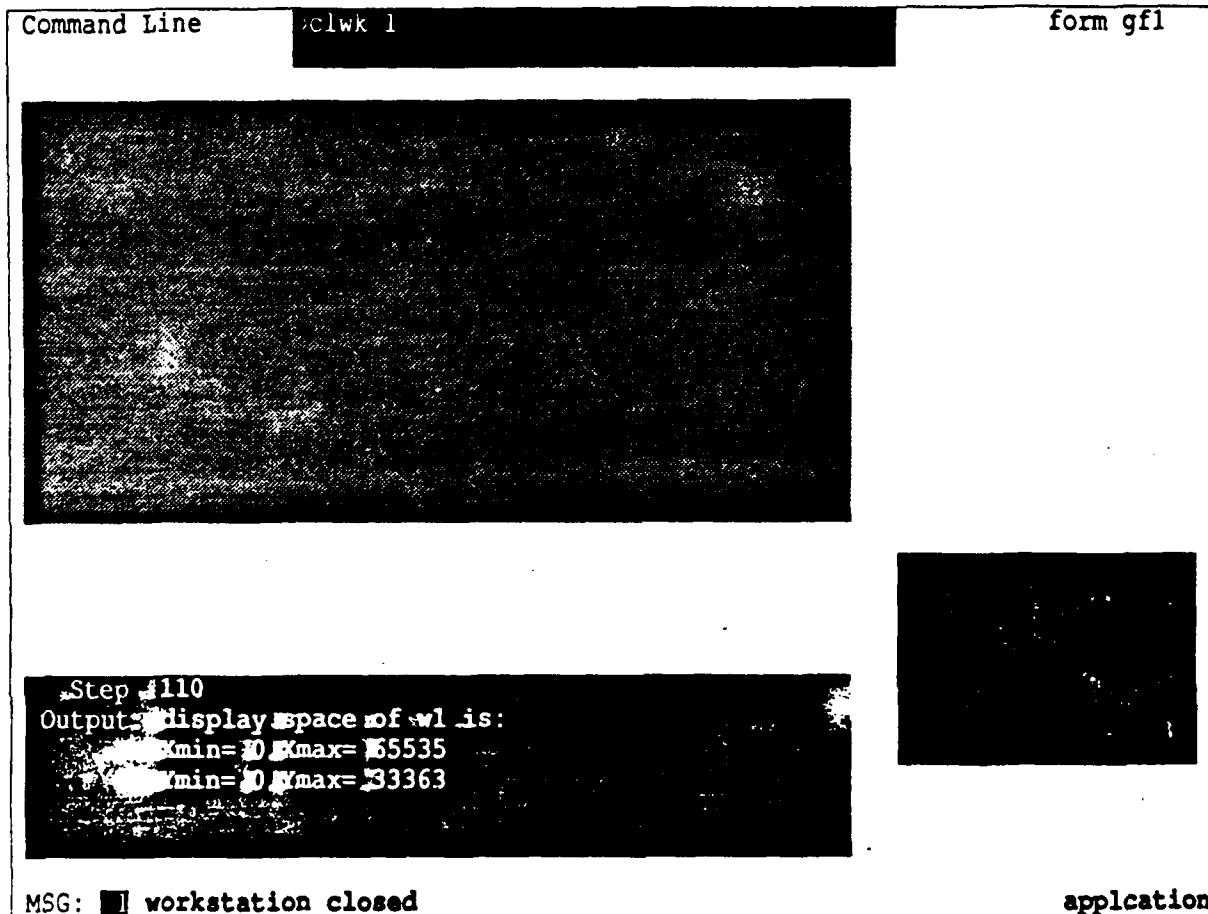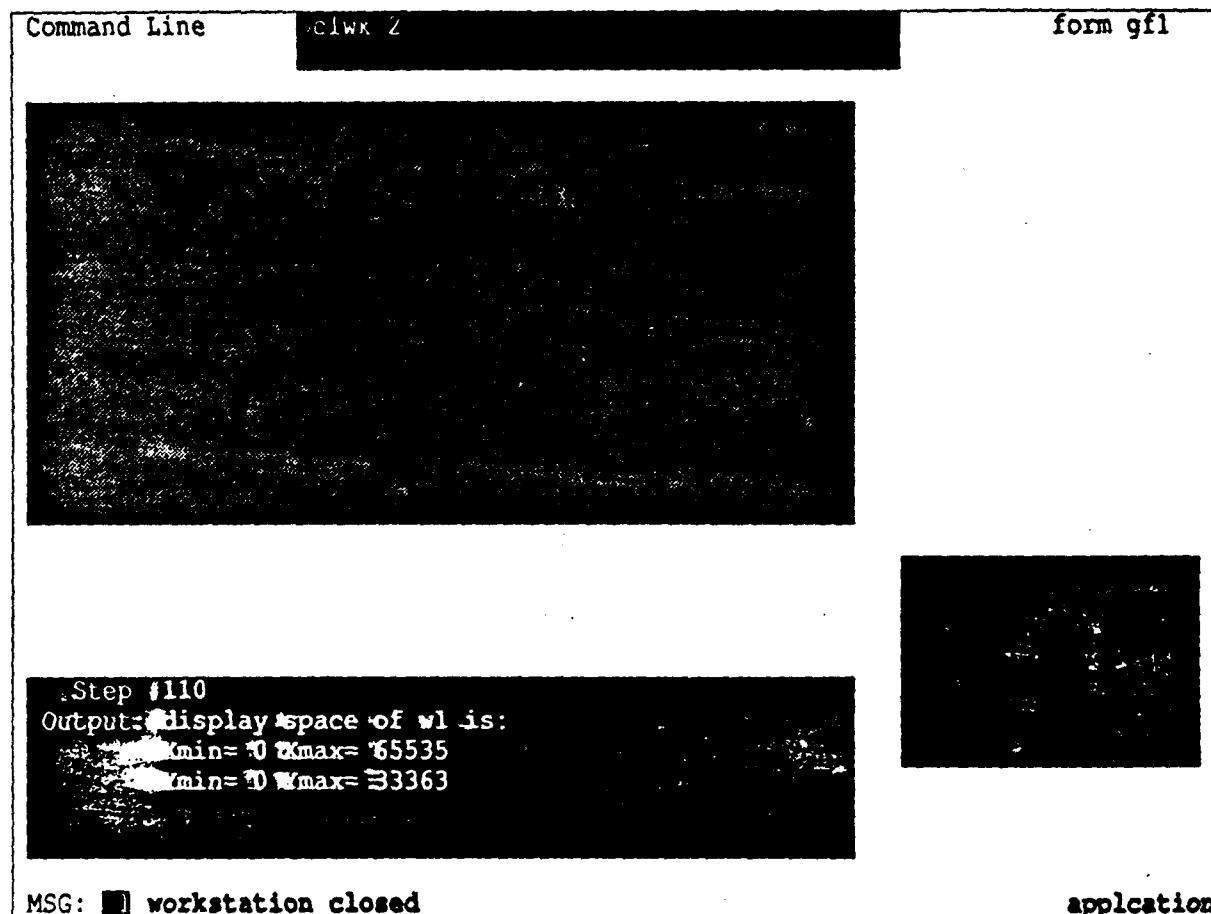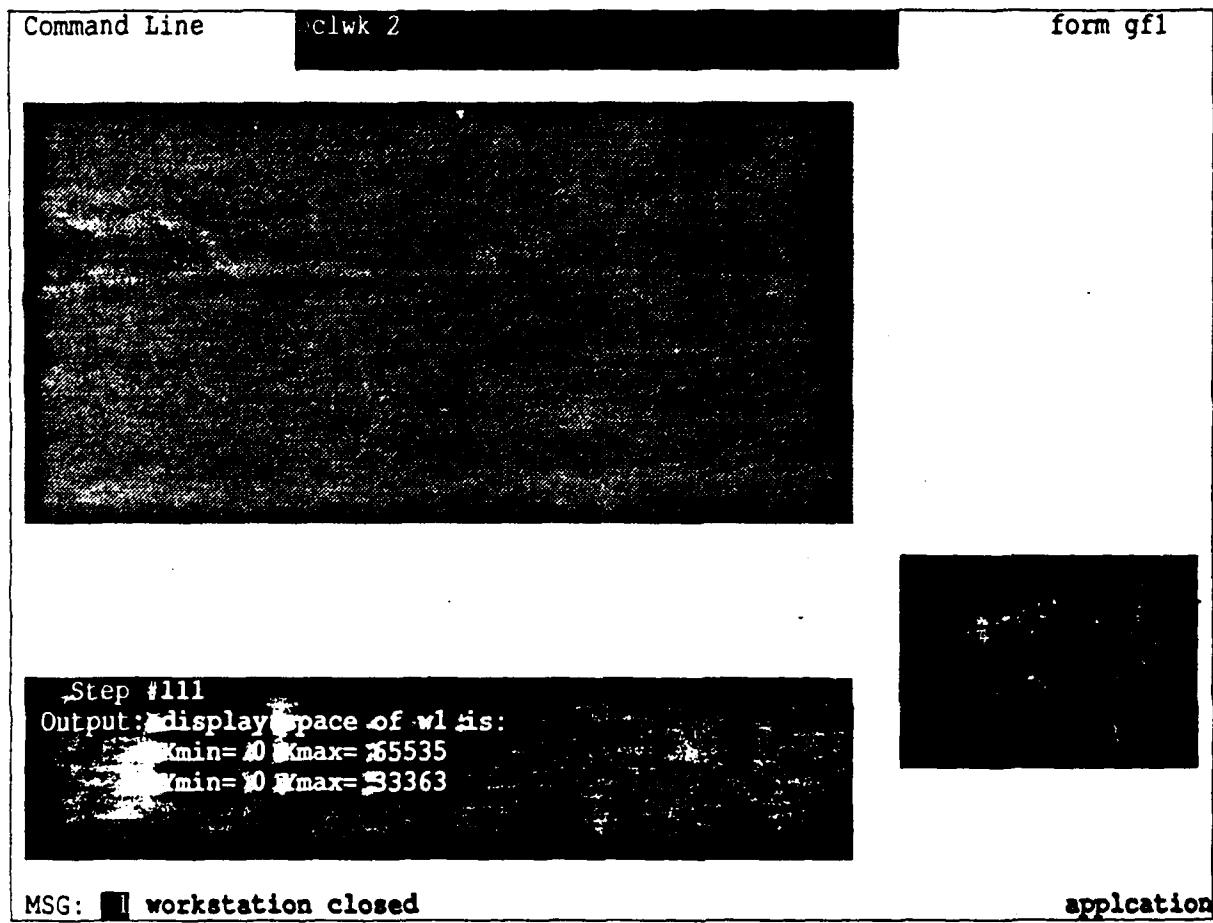
Figure 5-113b   (AFTER)

APPENDIX A

Commands for ARTEST

Commands are of the following form:

              `<command> <arg1> <arg2> ..... <argn>`

where the angle brackets are not actually entered.
`<command>` refers to the actual procedure call
`<arg1>` refers to argument 1, `<arg2>` to argument 2, etc.
arguments are separated by blank(s), and arguments which
contain blanks are enclosed in double quotes.

     To allow communication of transformation data, six
predefined matrices have been supplied. These are referred to
as 'matrix ID' in the calling argument string. It is the user's
responsibility to maintain the data in each. The names of
these are:

          2-dimensional: A2, B2, C2
          3-dimensional: A3, B3, C3

     The results from inquire functions and point transforms
will be shown in the "Display" output region, thus avoiding the
need for extra local variables.

     The available commands will be the short form as described
by the PHIGS C language binding. A description of these calls
follows:

1.   open PHIGS
     `<popph>`
2.   close PHIGS
     `<pclph>`
3.   open workstation
     `<popwk> <WS ID number> <path to window>`
        `<device name> <parent window ID> <page in window>`
4.   close workstation
     `<pclwk> <WS ID number>`
5.   set workstation window 2-D
     `<pswkw> <WS ID number> <min X> <max X> <min Y> <max Y>`
6.   set workstation window 3-D
     `<pswkw3> <WS ID number> <min X> <max X>`
                 `<min Y> <max Y> <min Z> <max Z>`
7.   set workstation viewport 2-D
     `<pswkv> <WS ID number> <min X> <max X> <min Y> <max Y>`
8.   set workstation viewport 3-D
     `<pswkv3> <WS ID number> <min X> <max X>`
                 `<min Y> <max Y> <min Z> <max Z>`

9.   scale 2-D
     `<psc> <scale X> <scale Y> <matrix ID>`
10. scale 3-D
     `<psc3> <scale X> <scale Y> <scale Z> <matrix ID>`
11. translate 2-D

```
             <ptr> <delta X> <delta Y> <predefined matrix ID>
12. translate 3-D
             <ptr3> <delta X> <delta Y> <delta Z> <matrix ID>
13. rotate about X axis
             <prox> <angle (degrees)> <matrix ID>
14. rotate about Y axis
             <proy> <angle> <matrix ID>
15. rotate about Z axis
             <proz> <angle> <matrix ID>
16. rotate 2-D
             <pro> <angle> <matrix ID>
17. compose matrix 2-D
             <pcom> <matrix ID 'A'> <matrix ID 'B'> <matrix ID (=AxB)>
18. compose matrix 3-D
             <pcom3> <matrix ID 'A'> <matrix ID 'B'> <matrix ID (AxB)>
19. transform point 2-D
             <ptp> <input X> <input Y> <matrix ID>
20. transform point 3-D
             <ptp3> <input X> <input Y> <input Z> <matrix ID>
21. build transformation matrix 2-D
             <pbltm> <ref X> <ref Y> <delta X> <delta Y>
                <scale X> <scale Y> <rotate angle> <matrix ID>
22. build transformation matrix 3-D
             <pbltm3> <ref X> <ref Y> <ref Z> <delta X> <delta Y>
                <delta Z> <scale X> <scale Y> <scale Z> <rotate X>
                   <rotate Y> <rotate Z> <result matrix ID>
23. compose transformation matrix 2-D
             <pcotm> <matrix ID> <ref X> <ref Y> <delta X> <delta Y>
                <scale X> <scale Y> <rotate angle> <resulting matrix ID>

24. compose transformation matrix 3-D
             <pcotm3> <matrix ID> <ref X> <ref Y> <ref Z> <delta X>
                <delta Y> <delta Z> <scale X> <scale Y> <scale Z>
                   <rotate X> <rotate Y> <rotate Z>
                      <resulting matrix ID>
25. message
             <pmsg> <message string>
26. error handler
             <perhnd> <error number> <function number>
27. error log
             <perlog> <error number> <function number>
28. set error handler mode
             <pserhm> <new mode (=ON, OFF)>
29. inquire error handler mode
             <pqerhm>

30. redraw all structures
             <prst> <workstation ID> <control (=CONDITIONALLY, ALWAYS)>
31. update workstation
             <puwk> <workstation ID> <regen flag (=PERFORM, POSTPONE)>
32. post structure
             <ppost> <workstation ID> <structure ID> <priority level>
33. unpost structure
             <pupost> <workstation ID> <structure ID>
34. unpost all structures
             <pupast> <workstation ID>
```

35. open structure
    <popst> <structure ID>
36. close structure
    <pclst>
37. delete structure
    <pdst> <structure ID>
38. delete structure network
    <pdsn> <structure ID> <ref flag (=DELETE, KEEP)>
39. delete all structures
    <pdas>

40. polyline 2-D
    <ppl> <num points> <X1> <Y1> <X2> <Y2> ... <Xn> <Yn>
41. polyline 3-D
    <ppl3> <num points> <X1> <Y1> <Z1> <X2> <Y2> <Z2> ....
          ... <Xn> <Yn> <Zn>
42. polymarker 2-D
    <ppm> <num points> <X1> <Y1> <X2> <Y2> ... <Xn> <Yn>
43. polymarker 3-D
    <ppm3> <num points> <X1> <Y1> <Z1> <X2> <Y2> <Z2> ....
          ... <Xn> <Yn> <Zn>
44. text 2-D
    <ptx> <X coord> <Y coord> <text string>
45. text 3-D
    <ptx3> <X> <Y> <Z> <Vec X-x> <Vec X-y> <Vec X-z>
              <Vec Y-x> <Vec Y-y> <Vec Y-z> <text string>
46. fill area 2-D
    <pfa> <num points> <X1> <Y1> <X2> <Y2> ... <Xn> <Yn>
47. fill area 3-D
    <pfa3> <num points> <X1> <Y1> <Z1> <X2> <Y2> <Z2> ...
          ... <Xn> <Yn> <Zn>
48. set local transform 2-D
    <pslmt> <matrix ID> <type (=PRE, POST, REPLACE)>
49. set local transform 3-D
    <pslmt3> <matrix ID> <type (=PRE, POST, REPLACE)>
50. set global transform 2-D
    <psgmt> <matrix ID>
51. set global transform 3-D
    <psgmt3> <matrix ID>
52. execute structure
    <pexst> <structure ID>

53. set polyline type
    <psln> <linetype (SOLID, DASH, DOT, DOTDASH)>
54. set polyline color index
    <psplci> <line color index (RED, WHITE, GREEN, etc.)>
55. set polymarker type
    <psmk> <marker type (POINT, PLUS, STAR, O, X)>
56. set polymarker color index
    <pspmci> <marker color index (RED, WHITE, GREEN, etc.)>
57. set character height
    <pschh> <character height>
58. set text color index
    <pstxci> <text color index (RED, WHITE, GREEN, etc.)>
59. set interior style

```
        <psis> <interior style (=HOLLOW, FILLED)>
60. set interior color index
        <psici> <color index (RED, WHITE, GREEN, etc.)>

61. inquire list of available workstation
        <pqewk>
62. inquire workstation connection and type
        <pqwkc> <workstation ID>
63. inquire display space size 2-D
        <pqdsp> <device name> <window ID> <page>
64. inquire display space size 3-D
        <pqdsp3> <device name> <window ID> <page>
```

65. in addition, the command "GRAPHICS" will be used to initialize the screen specifically designed for 3-D graphics. The command "FORM" may be used to return to the initial ARTEST setup screen.

```
        <graphics>
```